

AD-A086 766

L N K CORP SILVER SPRING MD

PATTERN RECOGNITION INVESTIGATION AND ANALYSIS OF PLANAR POINT --ETC (U)

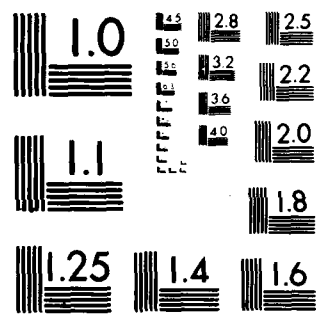
1979 D LAVINE, B LAMBIRD, L N KANAL

DAAK70-79-C-0091

NL

UNCLASSIFIED

2



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

5L
LEVEL II

(2)

Pattern Recognition Investigation and Analysis

or

Planar Point Patterns

ADA 086766

(12) 112

(10) L.N.K. Corp.
Enclosure/Lan and
under N/A

DTIC
ELECTE
JUL 16 1980
S D E

(9) Final Report

On

(15) Contract DAAK 70-79-C-0091

The views, opinions and/or findings contained in this report are those of L.N.K. Corporation and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

(11) 1979

L.N.K. Corporation

302 Notley Court

Silver Spring, MD 20904

391718 L.N.K.

80 6 11 076

DDC FILE COPY

Preface

This is the final report on work performed by L.N.K. Corporation under contract DAAK 70-79-C-0091. Dr. John W. Bond, Jr., DRDME-RT, U.S. Army Mobility Equipment Research and Development Command served as contract technical monitor. The report was prepared by L.N.K. scientists, Mr. David Lavine, Ms. Barbara Lambird and Dr. Laveen N. Kanal. Pattern description and reasons for performing this study are classified. The classified explanation can be obtained from Dr. Bond, telephone number (703) 664-4547, Autovon 354-5375 or 4547.

Accession For	
NTIS GRA&I <input checked="" type="checkbox"/>	
DDC TAB <input type="checkbox"/>	
Unannounced	
Justification <i>Handwritten</i>	
By _____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
<i>A</i>	

Summary

This is the final report on work performed by L.N.K. Corporation under contract DAAK 70-79-C-0091 from the U.S. Army Mobility Equipment Research and Development Command.

Two basic types of planar point patterns were investigated. The first type is specified by a set of points in a plane and a noise model which determines the allowed perturbations of the points subject only to the restriction that no interpoint distance should change by more than a certain percentage. The second type is specified by a set of points subject to angle and distance constraints.

Representations of point patterns having some invariance properties are presented. For the first type of prototype patterns, these include SIDV's ^{Vectors} Sorted Interpoint Distance Vector, SNN's - (Sorted Nearest Neighbor Vectors), and MST's (Minimal Spanning Trees). Theorems, experimental results of simulations, advantages, disadvantages and comparisons are presented for classification procedures based on these representations. It is concluded that classification of the first type of patterns can be accomplished using the methods based on SIDV's, SNN's and MST's under certain restrictions. If the number of points in the pattern is small, less than 100, then the SIDV method would be better, while more than 100, then the SNN or methods based on MST's would be more useful. If the percentage of additions or deletions of points is small, of the order of ^{100%} ~~ten percent~~, then the SDV, SNN, and MST are still feasible. If the percentage is large, then other methods must be used.

The techniques used to classify prototype point patterns give a measure of the quality of a match which can be used to provide a measure of the confidence on the classification of a particular pattern.

For the second type of point prototype specified by a set of constraints, only one example was provided. An algorithm that could recognize any instance of this pattern is presented.

Suggestions for further analytical and experimental investigation of classification schemes when the number of additions and deletions of pattern points becomes large are given in the final section of the report. Testing of all the devised classification schemes on actual noisy data from point patterns is also recommended.

<u>Table of Contents</u>	<u>Page</u>
Preface	1
Summary	11
1. Introduction	1-1
1.1 Basic Problem	1-1
1.2 Normal Observer and Planar Terrain	1-1
1.3 Non-Normal Observer and Planar Terrain	1-2
1.4 Non-Planar Terrain	1-2
1.5 Translation, Rotation, and Scaling of Point Patterns	1-2
2. Types of Patterns Investigated	2-1
2.1 Prototype Patterns	2-1
2.2 Pattern Specified by a Set of Constraints	2-1
3. Invariants of Prototype Patterns	3-1
3.1 Comparing Two Interpoint Distance Matrices	3-6
3.2 Canonical Forms of the Interpoint Distance Matrix	3-8
4. The Sorted Interpoint Distance Vector	4-1
4.1 Theory	4-1
4.2 Examples Using SIDV Algorithm	4-5
4.3 Experimentation Using SIDV Algorithm	4-10
5. Nearest Neighbor Algorithm	5-1
5.1 Theory	5-1
5.2 Example Using SNN Algorithm	5-6
5.3 Experimentation Using SNN Algorithm	5-9
(6. Minimal Spanning Tree Methods	6-1
6.1 Graph Theoretical Definitions of MST's	6-1

6.2	Theory of Using MST's for Classification	6-6
6.3	Examples of MST's	6-11
6.4	Experimentation with MST's	6-11
7.	Patterns Specified by a Set of Constraints	7-1
7.1	An Example	7-2
8.	Conclusions	8-1
9.	Recommendation for Future Investigation	9-1
9.1	MST	9-1
9.2	Additions and Deletions of Pattern Points	9-1
9.3	General Approach to Pattern Specified By Constraints	9-2
9.4	Testing of Classification Schemes	9-2
	Bibliography	10-1
	Appendix I	A-1
	Appendix 2	A-5
	Appendix 3	A-15
	Appendix 4	A-25

1. Introduction

This final report discusses the research performed by L.N.K. Corporation for MERADCOM's pattern investigation. The research included identifying important types of patterns, investigating invariants of the patterns, developing algorithms for finding these invariants, and finally developing and testing procedures for recognizing point patterns.

1.1 Basic Problem

It was assumed that extraction techniques for the point patterns had been successfully developed so that the patterns consisted of bright blobs embedded in a background of small to moderate noise. The noise appears in the patterns as relatively few blobs not as bright as the pattern blobs. The objective is: given a window with bright spots embedded in a background, recognize and classify the pattern. A formal statement of the problem is presented in Appendix I.

1.2 Normal Observer and Planar Terrain

At first, it is assumed that the points of the pattern occur in a flat or planar terrain and that the observation platform is normal to this plane. Patterns extracted under such conditions are referred to as ideal patterns. These patterns could have added points due to noise but no deleted points.

1.3 Non-Normal Observer and Planar Terrain

The terrain is still considered to be flat, but now the observation platform can be far from normal. All the points in the pattern are still visible, i.e. there is no fusion or coalescence of points. This case was not considered any different from the above case since there are standard techniques for correcting perspective distortions given the polar angle. For example see Sec 6.3 of Digital Picture Processing by Azriel Rosenfeld and Avnash C. Kak (Academic Press, 1976).

1.4 Non-Planar Terrain

If the terrain is not flat, then points can be masked from the observation platform due to the terrain, or points may coalesce. Using a terrain database, rectification of the image to the normal viewing angle is possible, but now the possibility of deletions of pattern points must be taken into account.

1.5 Translation, Rotation, and Scaling of Point Patterns

Since the observed patterns can have arbitrary orientation and location in the plane, the classification schemes should work regardless of any translations, rotations, or scaling. All reported classification schemes are invariant under rotation and translation. It is assumed that the altitude of the observation platform is known, so that scaling to a standard altitude is done on the point patterns before applying the recognition techniques.

2. Types of Patterns Investigated

Two basic types of point patterns were investigated. The first type of pattern is characterized by a prototype pattern which is specified by a set of points embedded in a plane and a noise model which determines the allowed perturbations. The second type of pattern is specified by a set of angle and distance constraints.

2.1 Prototype Patterns

Because actual samples of point patterns were generally unavailable, the emphasis of the investigation was on prototype patterns. These patterns are specified as a set of "ideal" patterns where each pattern is defined by a set of points embedded in a plane. The noise model then determines to what extent the points can move around in the plane. In this type of pattern, the number of points is considered fixed. Results of the investigation on the prototype patterns are given in Sections 3 to 6.

2.2 Pattern Specified by a Set of Constraints

This type of pattern is specified by a set of constraints. The constraints define regions specified by a set of angles, define the number of points possible in a region and can also limit the allowed interpoint distances. Thus, the number of points in such a pattern is not necessarily fixed.

One example of this type of pattern was made available to us and the results of investigating this pattern are discussed in Section 7. To investigate these types of patterns more generally, many more patterns need to be provided in order to be able to categorize the types of constraints that occur in such point patterns.

3. Invariants of Prototype Patterns

The basic approach used for matching each prototype pattern to its noisy versions is to define a way for representing both the prototype and noisy versions. These representations are then compared to determine whether or not they correspond to the same pattern. Some of the representations involve transformations that lose information so that it is not possible to reconstruct the original point pattern from the representation. Generally, the greater the loss of information in a representation, the simpler the representation and the easier it is to compare representations.

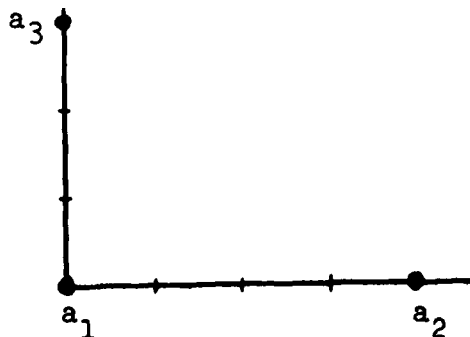
The most fundamental representation of a point pattern, which is invariant under rotation and translation, is the interpoint distance matrix. For a point set $A = \{a_1, \dots, a_N\}$ $a_i \in \mathbb{R}^2$, the interpoint distance matrix is defined to be the $N \times N$ matrix whose ij^{th} entry is the distance from point a_i to point a_j .

Ex. 3.1 Let $P = \{a_1, a_2, a_3\}$
be a point pattern where

$$a_1 = (0,0)$$

$$a_2 = (0,4)$$

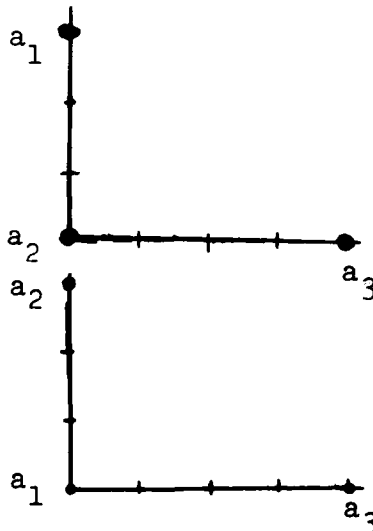
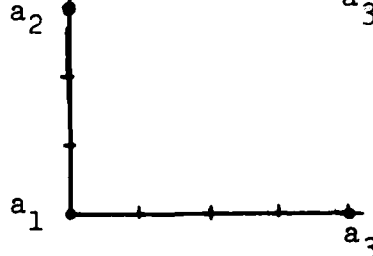
$$a_3 = (3,0)$$

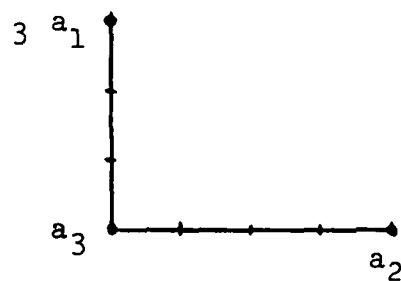


$$\text{Then } \text{IDM}(P) = \begin{pmatrix} 0 & 4 & 3 \\ 4 & 0 & 5 \\ 3 & 5 & 0 \end{pmatrix}$$

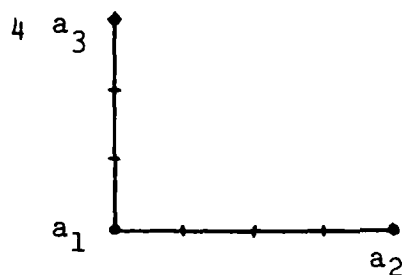
This representation is fundamental because it is possible to reconstruct the original point set from the matrix except for a translation, rotation and reflection. Since the interpoint distance matrix, IDM, and the original point set are equivalent in the above sense, it is natural to compare patterns by comparing their interpoint distance matrices. Unfortunately, many IDM'S can be derived from one point set, depending on the order in which the points are processed.

Ex. 3.2 Let $S = \{(0,0), (3,0), (0,4)\}$ be a three point pattern. There are six possible labellings of these points by labels a_1, a_2 and a_3 . We show these six labellings and the corresponding interpoint distance matrix whose ij^{th} entry is the distance from point i to point j .

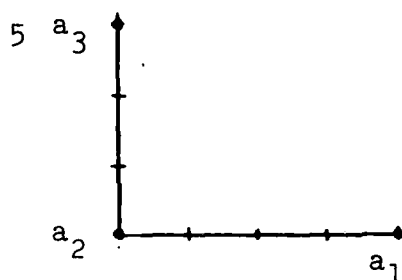
Case 1	Labelling	Interpoint Distance Matrix
		$\begin{pmatrix} 0 & 3 & 5 \\ 3 & 0 & 4 \\ 5 & 4 & 0 \end{pmatrix}$
2		$\begin{pmatrix} 0 & 3 & 4 \\ 3 & 0 & 5 \\ 4 & 5 & 0 \end{pmatrix}$



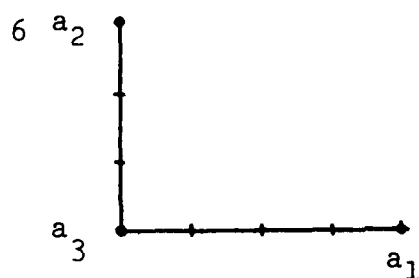
$$\begin{pmatrix} 0 & 5 & 3 \\ 5 & 0 & 4 \\ 3 & 4 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 4 & 3 \\ 4 & 0 & 5 \\ 3 & 5 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 4 & 5 \\ 4 & 0 & 3 \\ 5 & 3 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 5 & 4 \\ 5 & 0 & 3 \\ 4 & 3 & 0 \end{pmatrix}$$

The uncertainty in a IDM is determined up to a permutation of the entries in the matrix. Let $A = \begin{pmatrix} a_{11} \dots a_{1N} \\ a_{N1} \dots a_{NN} \end{pmatrix}$ be an interpoint distance matrix for pattern P. Let S_n be the set of all permutations of the integers $\{1, \dots, N\}$. Then the set of all interpoint distance matrices for the pattern P is given by

$$\left\{ \begin{pmatrix} a_{\pi(1)\pi(1)} & \dots & a_{\pi(1)\pi(N)} \\ \vdots & & \vdots \\ a_{\pi(N)\pi(1)} & \dots & a_{\pi(N)\pi(N)} \end{pmatrix} \mid \pi \in S_n \right\}$$

Ex. 3.3 Let $S_n = \{\pi_1, \dots, \pi_6\}$ where

$$\pi_1 : \begin{pmatrix} 1 \rightarrow 1 \\ 2 \rightarrow 2 \\ 3 \rightarrow 3 \end{pmatrix} \quad \pi_4 : \begin{pmatrix} 1 \rightarrow 1 \\ 2 \rightarrow 3 \\ 3 \rightarrow 2 \end{pmatrix}$$

$$\pi_2 : \begin{pmatrix} 1 \rightarrow 2 \\ 2 \rightarrow 1 \\ 3 \rightarrow 3 \end{pmatrix} \quad \pi_5 : \begin{pmatrix} 1 \rightarrow 2 \\ 2 \rightarrow 3 \\ 3 \rightarrow 1 \end{pmatrix}$$

$$\pi_3 : \begin{pmatrix} 1 \rightarrow 3 \\ 2 \rightarrow 1 \\ 3 \rightarrow 2 \end{pmatrix} \quad \pi_6 : \begin{pmatrix} 1 \rightarrow 3 \\ 2 \rightarrow 2 \\ 3 \rightarrow 1 \end{pmatrix}$$

Let $a_1 = (0, 4)$ $a_2 = (3, 0)$ and $a_3 = (0, 0)$ (case 3)

As in example 3.2, the IDM is

$$\begin{pmatrix} 0 & 5 & 3 \\ 5 & 0 & 4 \\ 3 & 4 & 0 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

The permuted IDM corresponding to π_2 is

$$\begin{pmatrix} a_{\pi_2(1)\pi_2(1)} & a_{\pi_2(1)\pi_2(2)} & a_{\pi_2(1)\pi_2(3)} \\ a_{\pi_2(2)\pi_2(1)} & a_{\pi_2(2)\pi_2(2)} & a_{\pi_2(2)\pi_2(3)} \\ a_{\pi_2(3)\pi_2(1)} & a_{\pi_2(3)\pi_2(2)} & a_{\pi_2(3)\pi_2(3)} \end{pmatrix}$$

$$= \begin{pmatrix} a_{22} & a_{21} & a_{23} \\ a_{12} & a_{11} & a_{13} \\ a_{32} & a_{31} & a_{33} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 5 & 4 \\ 5 & 0 & 3 \\ 4 & 3 & 0 \end{pmatrix}$$

Note that this is just case 6 in example 3.2

3.1 Comparing Two Interpoint Distance Matrices

Since the set S_n contains $n!$ elements, the problem of comparing two IDM's can be formidable.

In the simplest possible comparison, we are given two interpoint distance matrices and ask if they correspond to the same unperturbed point set, which involves comparing $n!$ permutations.

A more realistic and complex version of the problem involves comparing two IDM's determined from two different point patterns. The problem is to decide if one of the point patterns is a perturbation of the other. The comparison is done by finding the permutation of the best component by component fit with the entries of the other matrix. This search might be performed by doing a least squares fit of one matrix with each permutation of the other matrix and selecting the permutation yielding the smallest sum of least square values. Since this procedure is computationally expensive, other algorithms were sought to provide approximate solutions to the problem.

Ex. 3.4 Consider the pattern $R = \{ (0,0), (3.1,0), (0,4.2) \}$

The interpoint distance matrix is

$$A = \begin{pmatrix} 0 & 3.1 & 5.2 \\ 3.1 & 0 & 4.2 \\ 5.2 & 4.2 & 0 \end{pmatrix}$$

Let A and B be 3×3 matrices whose ij^{th} components are a_{ij} and b_{ij} respectively. Define $d(A,B)$, the distance between A and B by

$$d(A,B) = \sqrt{\sum_{j=1}^3 \sum_{i=1}^3 (A_{ij} - B_{ij})^2}$$

3-6

To determine the closeness of pattern R and pattern S of Ex. 3.2, we compute $d(A, S_i)$ for $i = 1, \dots, 6$ where the S_i are the IDM's of S. We see that A is closest to case 1 and the distance is $.1^2 + .2^2 + .2^2 = .09$.

3.2 Canonical Forms of the Interpoint Distance Matrix

One approach to finding approximate pattern matches is to compute new representations from the IDM's and compare the patterns by using the new representations. These representations are invariant under permutations in S_n . This means two interpoint distance matrices, which differ only by a permutation belonging to S_n , will produce the same representation.

One such class of representations is a polynomial on the entries of the matrix which does not change in value if the matrix is permuted. A simple example of such a polynomial is the product of all the non-diagonal entries in the matrix. The construction of polynomials defined on R^n and invariant under certain types of permutations has been extensively studied [1].

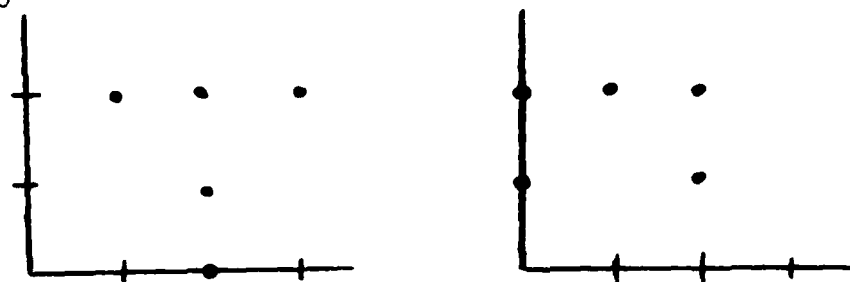
Ex. 3.5 Let $S = \{(0,0), (3,0), (0,4)\}$ be the pattern in example 3.2. In each of the six IDM's for this pattern the product of the non-diagonal elements is $3 \cdot 3 \cdot 4 \cdot 4 \cdot 5 \cdot 5 = 3,600$. Note that this number is independent of the choice of IDM's.

Several problems arise with the use of invariant polynomials. First, such invariant polynomials are non-trivial to find. Second, the invariant polynomials involve multiplication of interpoint distances, so it is impossible to determine how much the individual components contributed.

To avoid these difficulties we sought a different type of invariant. To each IDM we associate the vector whose entries are the elements, $\{a_{ij}\}$ $j > i$, of the matrix but sorted in increasing order. We denote this sorted interpoint distance vector by SIDV.

A SIDV is invariant under permutations of the IDM from which it is derived. In the following example the two inequivalent IDM's give rise to the same SIDV. We have been unable to determine if this is true for point patterns with more than five points.

Ex. 3-6



The usefulness of the SIDV is based on the following observations:

- 1) In practice we have not encountered distinct patterns with nearly identical SIDV's.
- 2) The SIDV is completely determined by the set of points and not upon the order in which they are processed.
- 3) Noisy versions of a pattern have SIDV's similar to that of the original pattern.
- 4) SIDV's can be used to define a similarity measure between patterns.

Before elaboration on the above points, we mention a problem with this approach. The comparison of two SIDV's becomes very expensive if the cardinalities of the corresponding point sets differ.

4. The Sorted Interpoint Distance Vector

The SIDV was useful for several classification algorithms. These procedures provide a measure of the quality of a match between a noisy pattern and each ideal pattern. Initially, we assume that a noisy pattern is obtained by perturbing points in the ideal pattern, so points are neither added nor deleted. Later, this restriction is lessened slightly, but a large number of additions or deletions will require other types of algorithms.

4.1 Theory

Def. Let $P = \{a_1, \dots, a_n\}$ be a pattern. The interpoint distance matrix, $IDM(P)$, of P is defined to be the $n \times n$ matrix whose ij^{th} entry is the Euclidean distance from a_i to a_j .

Def. The sorted interpoint distance vector, SIDV, of the pattern $P = \{a_1, \dots, a_n\}$ is the vector of size $\frac{n^2-n}{2}$ whose entries are the elements $\{IDM(P)_{ij}\}$ in increasing order where $IDM(P)_{ij}$ is the ij^{th} entry of $IDM(P)$.

Ex. 4.1 Let $S = \{(0,0), (3,0), (0,4)\}$ be the pattern of example 3.2. Its interpoint distance matrix is:

$$\begin{pmatrix} 0 & 3 & 4 \\ 3 & 0 & 5 \\ 4 & 5 & 0 \end{pmatrix}$$

The entries above the diagonal are 3, 4, and 5.
Thus these elements, sorted in increasing order
give:

$$\text{SIDV}(S) = (3, 4, 5).$$

Our first point pattern classification procedure SIDV compares a pattern P with an ideal pattern P_1 by comparing their vectors $\text{SIDV}(P)$ and $\text{SIDV}(P_1)$ componentwise. P and P_1 are said to be an α -percent match if

$$\frac{|\text{SIDV}(P_1)_j - \text{SIDV}(P)_j|}{\text{SIDV}(P_1)_j} \leq \alpha$$

for all j , where the subscript j denotes the j^{th} component of the vector. We now show that SIDV is an α -percent regular point pattern classification procedure. (See Appendix I for the definition of an α -percent regular point pattern classification procedure.) This is proved in Theorem I.

This theorem guarantees that if we take a sorted list of numbers, perturb each by no more than α -percent and sort this list then it must match the original list component by component to within α -percent. Thus if the first list is the SIDV for a pattern and the second is the SIDV for an α -percent perturbation, then the lists must match component by component to within α -percent.

Thm I Let $A = \{a_1, \dots, a_m\}$ and $B = \{b_1, \dots, b_m\}$ where $a_i \in R$, $b_i \in R$ for $i=1, \dots, m$ and $a_1 \leq a_2 \leq \dots \leq a_m$ and $b_1 \leq b_2 \leq \dots \leq b_m$. Let $M = \{1, \dots, m\}$ and assume there exists a 1-1 and onto map $f: M \rightarrow M$ such that

$$\left| \frac{a_1 - b_{f(i)}}{a_1} \right| \leq \alpha \quad \text{for } i = 1, \dots, m$$

Then
$$\left| \frac{a_1 - b_1}{a_1} \right| \leq \alpha \quad \text{for } i = 1, \dots, m.$$

Proof: If $f(i) = i$ for all $1 \leq i \leq m$ then we are done. Thus we are reduced to showing that the result holds for any i_0 such that $f^{-1}(i_0) < i_0$ or $f^{-1}(i_0) > i_0$.

Case 1) $c = f^{-1}(i_0) < i_0$.

$$\text{Hence } a_c \leq a_{i_0}.$$

$$\text{Thus } b_{i_0} \leq (1 + \alpha) a_c \leq (1 + \alpha) a_{i_0}.$$

Subcase 1) $g = f(i_0) < i_0$

$$b_{i_0} \geq b_g \geq (1 - \alpha) a_{i_0}$$

Subcase 2) $g = f(i_0) > i_0$

$$b_{i_0} \leq b_g \leq (1 + \alpha) a_{i_0}$$

Claim: there exists a $j_0 > i_0$ such that

$$f(j_0) < i_0.$$

$$\text{From this we see } b_{i_0} \geq b_{f(j_0)} \geq$$

$$(1 - \alpha) a_{j_0} \geq (1 - \alpha) a_{i_0}.$$

The claim follows from the fact that if $f(k) > i_0$ for $k \in \{i_0 + 1, \dots, m\}$ then since $f(i_0) > i_0$ we must have $f(k_1) = f(k_2)$ for some $k_1, k_2 \in \{i_0 + 1, \dots, m\}$. This contradicts the assumption that f is 1-1.

Case 2) $f^{-1}(i_0) > i_0$.

The proof is analogous to that of case 1.

This fact allows a measure of the quality of the fit to be made. If any pair of corresponding components fail to match to within α -percent then P can not be an α -percent noisy pattern of P_1 . Any of the standard measures of distance between vectors, such as Euclidean distance, can be computed, resulting in a measure of the goodness of fit.

4.2 Example Using SIDV Algorithm

In this section, we demonstrate how SIDV is used to classify patterns. Fig 4-1(a) shows an ideal pattern (solid circles) called A and an α -percent noisy version of that pattern (open circles) called N. Fig 4-1(b) shows a different ideal pattern, called B.

The interpoint distances of each of the three patterns are calculated and sorted. A component by component difference is then computed. Fig 4-2(a) shows ideal pattern A compared with noisy pattern N. The third column is the absolute difference between their sorted interpoint distance vectors. Fig 4-2(b) shows ideal pattern B similarly compared with noisy pattern N.

Suppose a threshold for the difference is selected and the number of differences greater than this threshold is counted. If a threshold of 2.0 is picked, then no differences greater than 2.0 occur in the comparison of pattern A and N. But, five differences are greater than 2.0 in the comparison of patterns B and N.

Using a threshold of 2.0, pattern N would be classified as an α -percent noisy version of pattern A and not pattern B. Fig 4-3 shows a plot of the total number of differences greater than the threshold of 2.0.

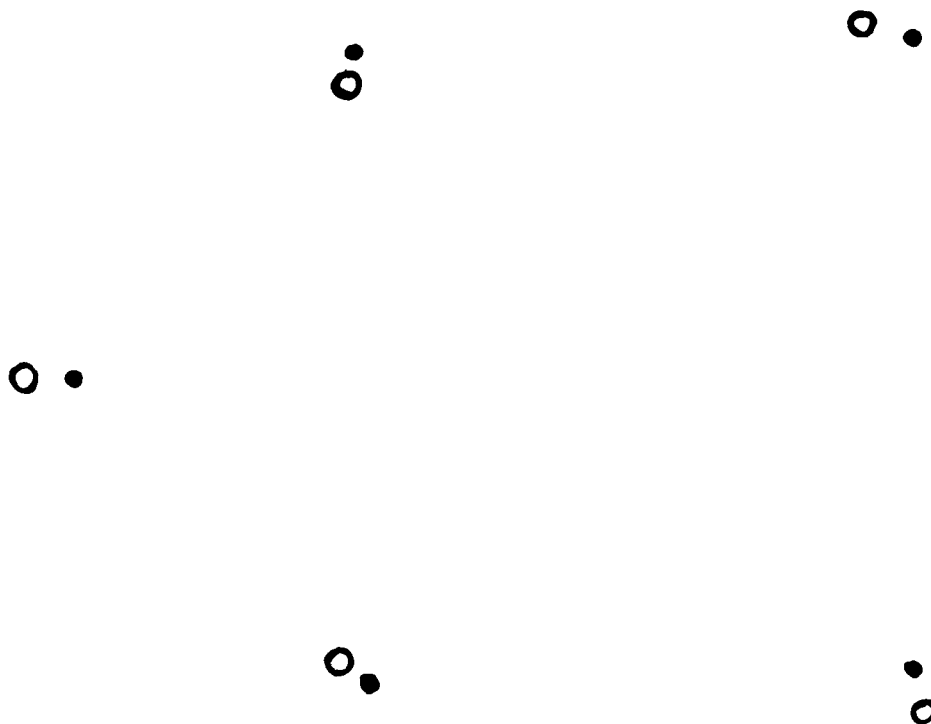


Fig 4-1(a). The solid circles form ideal pattern A. The open circles form pattern N, an α -percent noisy version of pattern A.



Fig. 4-1(b). The solid circles form ideal pattern B.

Sorted Interpoint Distances

<u>Ideal A</u>	<u>Noisy N</u>	<u>Difference</u>	<u>Diff Threshold>2</u>
6.2	6.0	0.2	
6.2	6.3	0.1	
8.0	7.5	0.5	
8.0	8.2	0.2	
8.0	8.6	0.6	
8.0	9.9	1.9	
12.1	11.9	0.2	
12.1	12.2	0.1	
13.1	13.1	0.0	
13.1	13.9	0.8	

Fig 4-2(a). Comparing SIDV's of patterns A and N.

<u>Ideal B</u>	<u>Noisy N</u>	<u>Difference</u>	<u>Diff Threshold>2</u>
6.2	6.0	0.2	
6.2	6.3	0.1	
8.0	7.5	0.5	
8.0	8.2	0.2	
9.0	8.6	0.4	
14.2	9.9	4.3	✓
14.2	11.9	2.3	✓
15.8	12.2	3.6	✓
15.8	13.1	2.7	✓
20.6	13.9	6.7	✓

Fig 4-2(b). Comparing SIDV's of patterns B and N.

Example of Using
SIDV

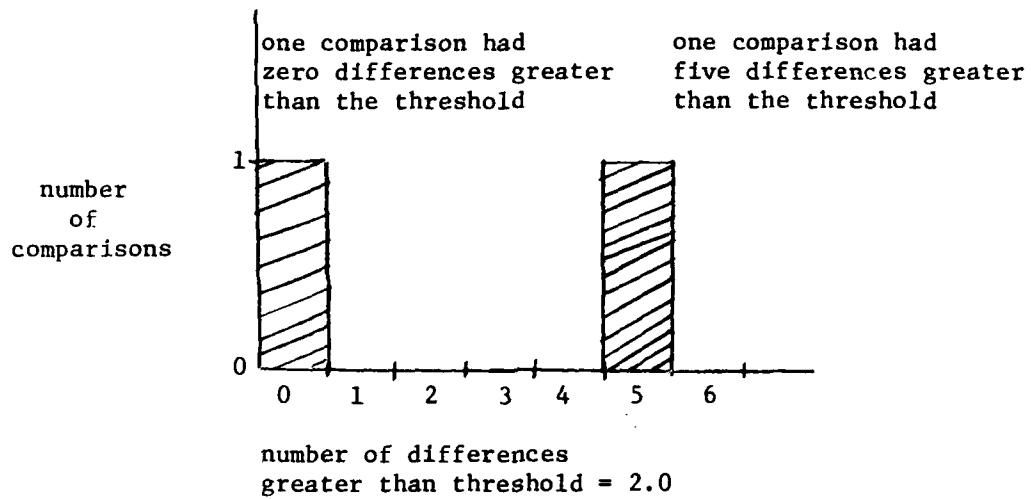


Fig 4-3. Histogram of Result of Comparing Noisy Pattern N with Ideal Patterns A and B.

4.3 Experimentation Using SIDV Algorithm

Three patterns each with twenty points were created, shown in Fig 4-4(a), (b), and (c), to be used as ideal patterns. Uniform noise within a rectangle about each point was used to create three noisy versions of each of the three ideal patterns, as demonstrated on one point in Fig 4-4(a).

Each of the noisy versions was compared with each of the ideal patterns as explained in Sec 4.2. The absolute difference between corresponding components ranged from 0 to 20. A wide range of thresholds were possible for perfect classification. Using an absolute difference threshold of 4.0, the results are shown in Fig 4-5. The x-axis shown the total number of differences greater than the threshold. Note that the cluster of comparing ideal patterns with noisy versions of themselves is well separated from the cluster of comparing ideal patterns with noisy versions of other ideal patterns.

The effect of deleting a point from a noisy version of an ideal pattern was investigated next. Deleted points are possible when dealing with non-planar terrain as discussed in Sec 1.4.

For each of the three ideal patterns, five noisy patterns were generated but with one point missing (a different point each time). The SIDV's of the noisy patterns are shorter than the SIDV's of the ideal patterns since the patterns have fewer points.

In order to compare using corresponding component differences, the two sets of SIDV's must be made the same length. This was done by adding distances to the shorter SIDV's. The added distances were uniformly distributed over the range of values in the shorter list. (If a point on the outskirts of an ideal pattern was deleted,

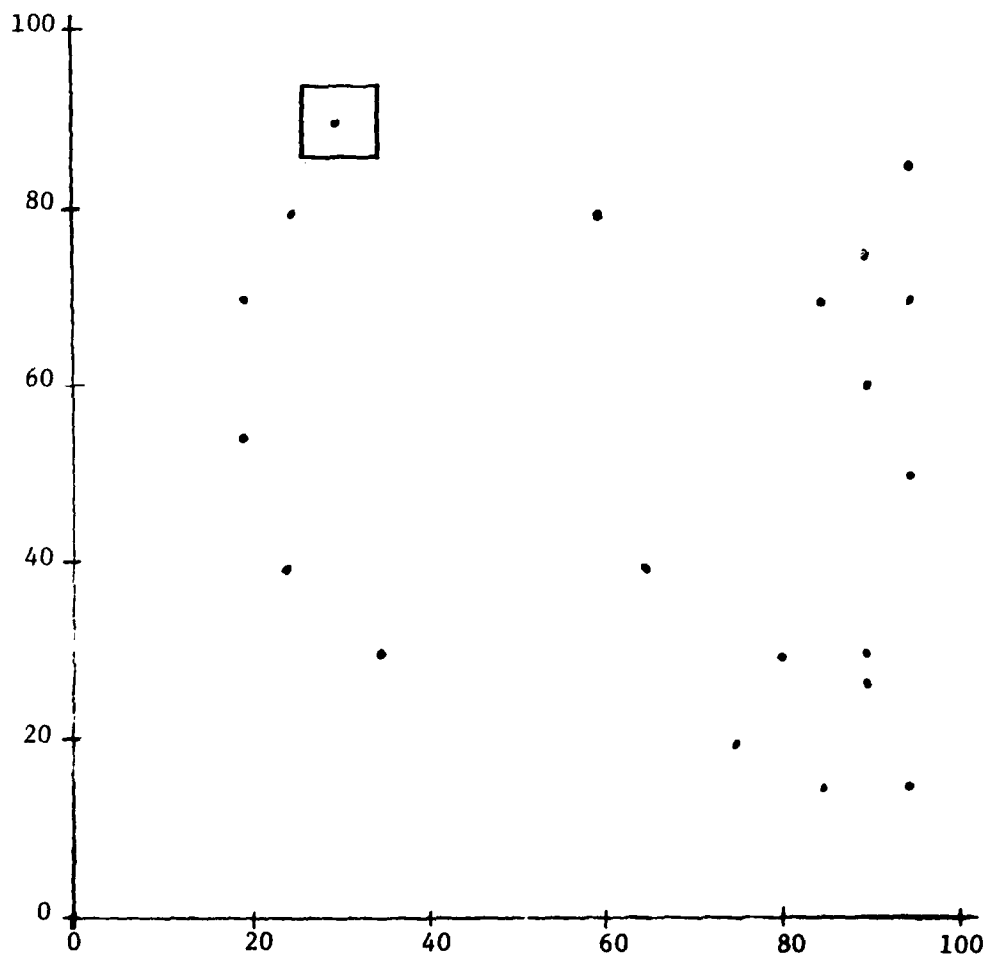


Fig 4-4(a). Synthetic pattern with twenty points used in testing the SIDV classification procedure. The box around the point represents the amount of noise allowed for each point.

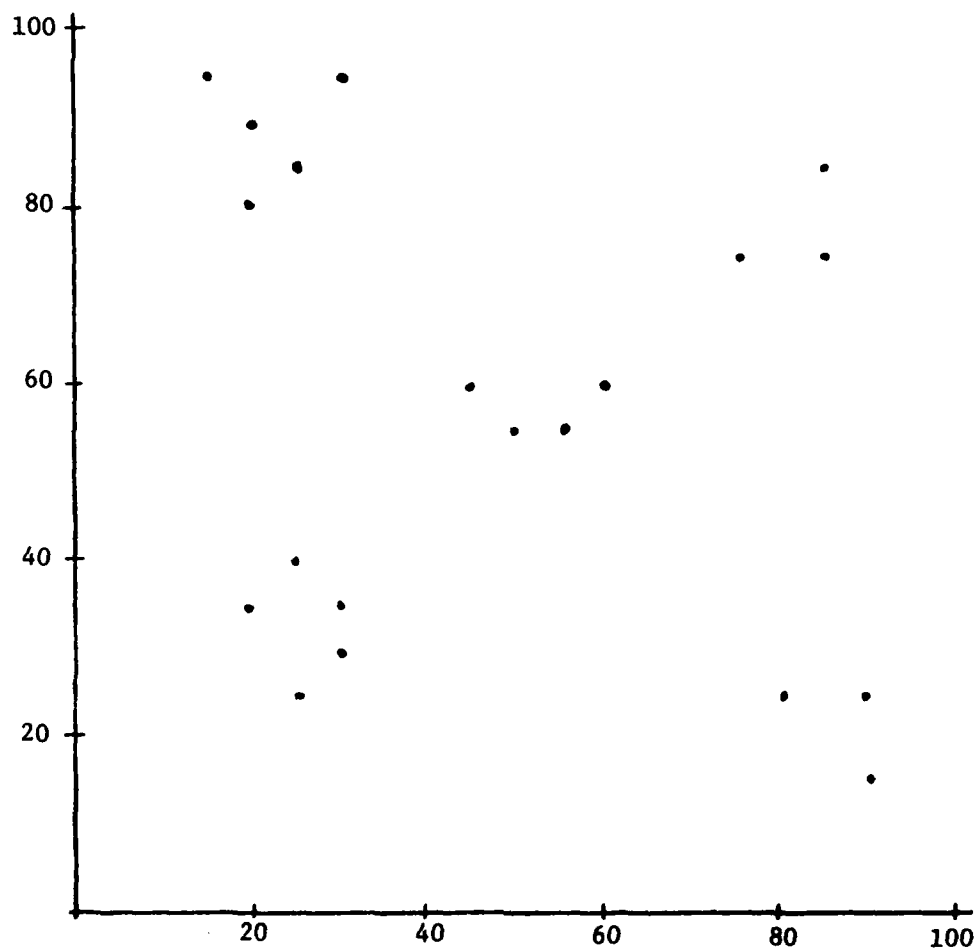
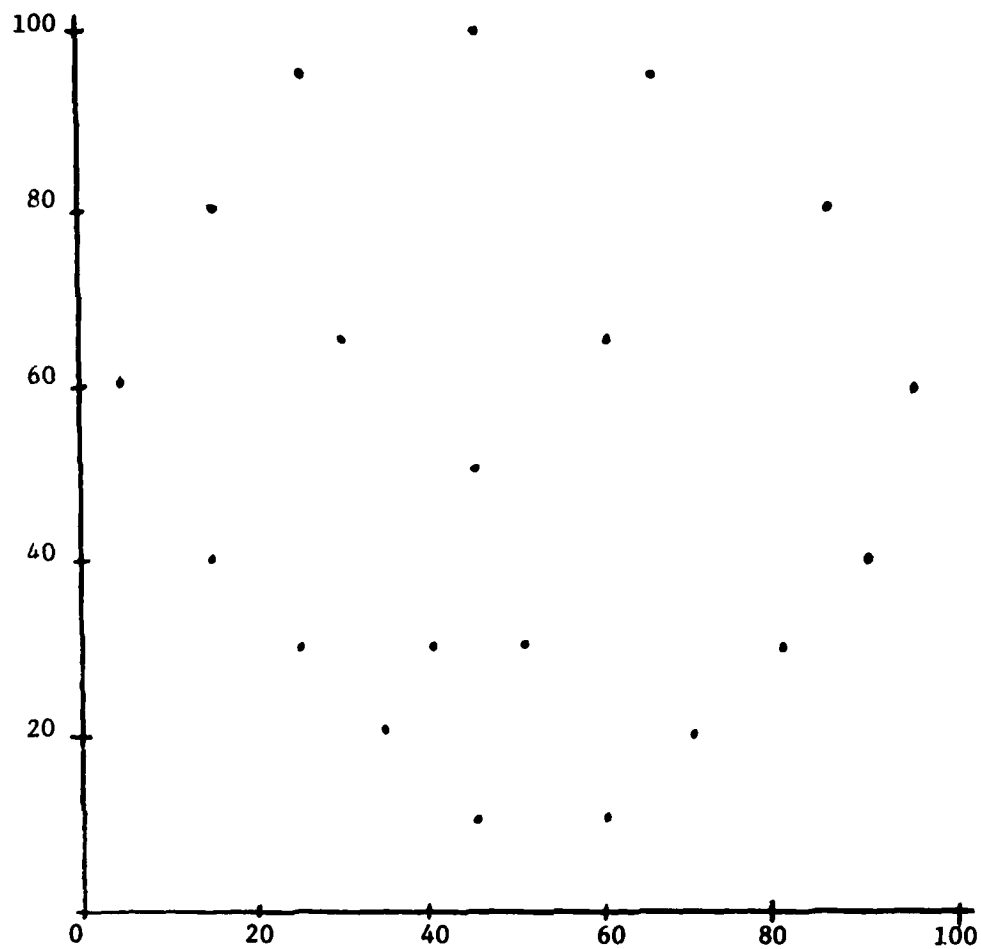


Fig 4-4(b). Synthetic pattern used for testing the SIDV classification procedure.



(Fig 4-4(c). Synthetic pattern used for testing the SIDV classification procedure.

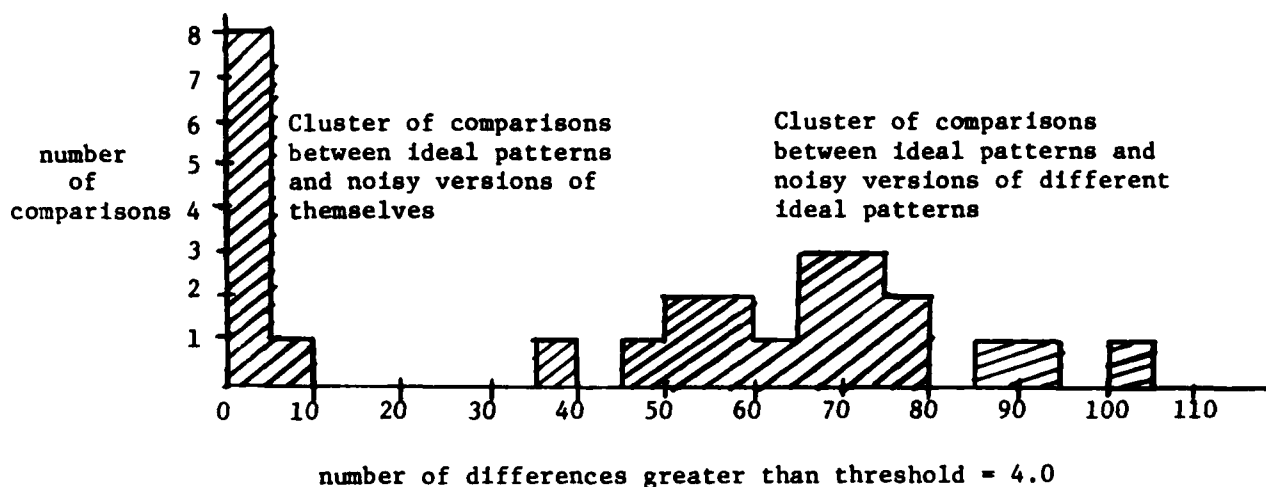


Fig 4-5. Histogram of result of experimentation comparing three synthetic patterns with noisy versions of themselves. The three patterns are shown in Fig 4.4.

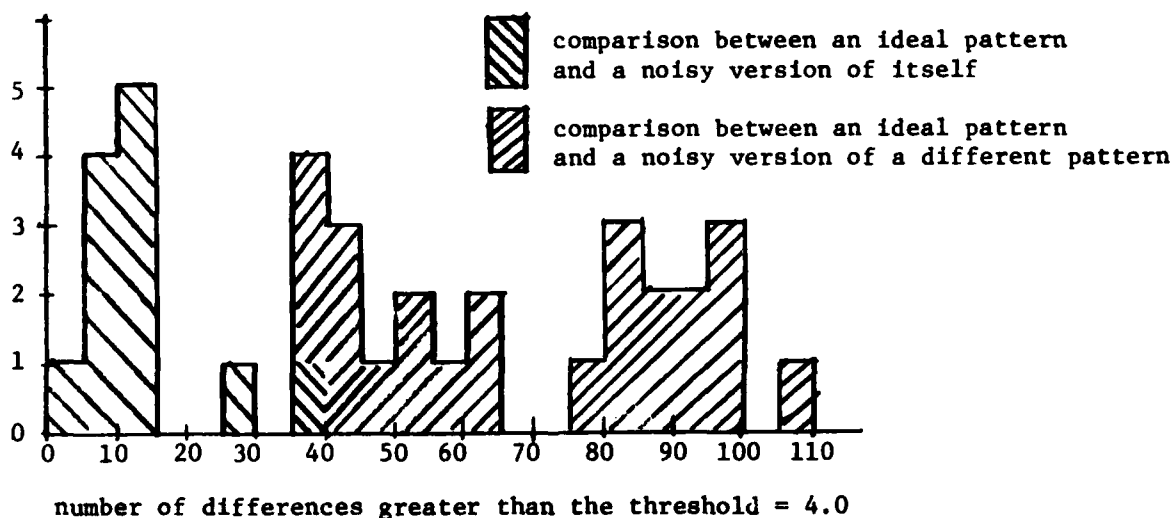


Fig 4-6. Histogram of result of experimentation comparing three synthetic patterns with five noisy versions of themselves. The noisy patterns had one deleted point.

the missing distances would range over the possible distance lengths. Thus adding distances that are distributed over the range of possible distance lengths is like adding a point to the edge of the ideal pattern.)

Using the threshold of 4.0 it was possible to correctly classify thirty-four out of the thirty-six comparisons. The histogram is shown in Fig 4-6.

5. Nearest Neighbor Algorithm

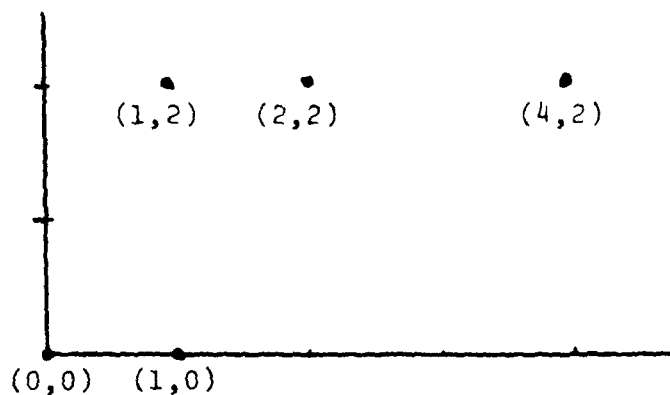
The various classification procedures using the sorted interpoint distance vector provided good classification in limited experimentation, but did not appear feasible if noisy patterns contained more than one or two additions or deletions. The primary problem in this more general situation is that SIDV methods involve matching a sequence to a subsequence of another sequence, which can rapidly become computationally expensive. To overcome the combinatorial problem, a nearest neighbor procedure was devised where a pattern consisting of n points is represented by a vector with n components in contrast to the $\frac{n(n-1)}{2}$ components required by the SIDV methods.

5.1 Theory

The vector of length n assigned to a pattern by the nearest neighbor procedure consists of finding the distance from each point to its closest neighbor and then sorting these distances in increasing order.

Def. Let $P = \{a_1, \dots, a_n\}$ be a pattern. The sorted nearest neighbor vector, denoted $SNN(P)$, is the n -tuple whose i^{th} component denoted $SNN(P)_i$ is the i^{th} largest element in the collection $\min_{i \neq j} d \{(a_i, a_j)\}_{i=1}^n$.

Ex. 5.1



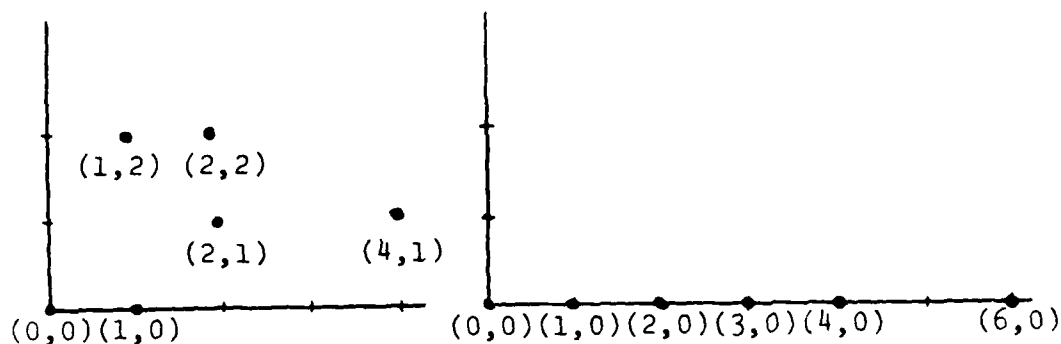
<u>Point</u>	<u>Nearest Neighbor</u>	<u>Distance</u>
(0,0)	(1,0)	1
(1,0)	(0,0)	1
(1,2)	(2,2)	1
(2,2)	(1,2)	1
(4,2)	(2,2)	2

$SNN = (1, 1, 1, 1, 2)$

The motivations for selecting this vector $SNN(P)$ to represent the pattern P are: 1) the size of the vector is small, reducing combinatorial problems in matching, 2) an α -percent regular point classification procedure can be defined using $SNN(P)$. Major disadvantages of this scheme are 1) many dissimilar patterns have identical sorted nearest neighbor vectors, 2) the sorted nearest neighbors are mainly the relatively shorter edge lengths in the complete graph of the pattern and short edge

lengths may tend to vary by a longer fraction of their length than longer ones.

Ex. 5.2



The two patterns shown above have the same

$$\text{SNN} = (1,1,1,1,1,2)$$

The SNN algorithms are identical to the SIDV algorithms except that, of course, the SNN is used. As a first step in understanding the usefulness of the SNN, we examine its behavior under α -percent perturbations.

Lemma I Let $P = \{a_1, \dots, a_n\}$ be a pattern and let

$P' = \{b_1, \dots, b_n\}$ be an α -percent perturbation of P

such that b_i corresponds to a_i for $i = 1, \dots, n$. For

$i = 1, \dots, n$ let $d_i = \min_{\substack{j \neq i \\ 1 \leq j \leq n}} d(a_i, a_j)$ and let

$d'_i = \min_{\substack{j \neq i \\ 1 \leq j \leq n}} d(b_i, b_j)$. Then $(1-\alpha)d_i \leq d'_i \leq (1+\alpha)d_i$ for

$i = 1, \dots, n$.

Proof: Let $1 \leq i_0 \leq n$. Select j_0 such that $d(b_{i_0}, b_{j_0}) = d'_{i_0}$

and k_0 such that $d(a_{i_0}, a_{k_0}) = d_{i_0}$, where

$$d_{i_0}' = d(b_{i_0}, b_{j_0}) \geq (1-\alpha)d(a_{i_0}, a_{j_0}) \geq (1-\alpha)d(a_{i_0}, a_{k_0}) \\ \geq (1-\alpha)d_{i_0}$$

and

$$d_{i_0}' = d(b_{i_0}, b_{j_0}) \leq d(b_{i_0}, b_{k_0}) \leq (1+\alpha)d(a_{i_0}, a_{k_0}) \\ \leq (1+\alpha)d_{i_0}.$$

$$\text{Thus } (1-\alpha)d_{i_0} \leq d_{j_0}' \leq (1+\alpha)d_{i_0}.$$

So the distance of a point to its nearest neighbor can change by no more than α -percent under an α -percent perturbation. Using Lemma I and Theorem I, we show that the SNN changes componentwise by no more than α -percent under an α -percent perturbation.

Thm II Let $P = \{a_1, \dots, a_n\}$ be a pattern and let $P' = \{b_1, \dots, b_n\}$ be an α -percent perturbation of P . Then

$$\frac{|SNN(P)_i - SNN(P')_i|}{SNN(P)_i} \leq \alpha$$

for $i = 1, \dots, n$.

Proof: We first show that if $v, w \in P$ and w is the nearest neighbor of v and v' and w' are the corresponding points in P' , and $z' \in P'$ is a nearest neighbor of v' , then

$$\frac{|d(v, w) - d(v', z')|}{d(v, w)} \leq \alpha$$

Since P' is an α -percent perturbation of P ,

$$(1-\alpha)d(v,w) \leq d(v',w') \leq (1+\alpha)d(v,w).$$

Thus if z' is the nearest neighbor of v' ,

$$d(v',z') \leq d(v',w') \leq (1+\alpha)d(v,w).$$

Assume z is the point in P corresponding to z' in P' .

Then

$$(1-\alpha)d(v,z) \leq d(v',z') \leq (1+\alpha)d(v,z).$$

Since w is the nearest neighbor of v ,

$$d(v,w) \leq d(v,z).$$

Thus

$$(1-\alpha)d(v,w) \leq (1-\alpha)d(v,z) \leq d(v',z').$$

We now have

$$(1-\alpha)d(v,w) \leq d(v',z') \leq (1+\alpha)d(v,w).$$

Hence

$$\left| \frac{d(v,w) - d(v',z')}{d(v,w)} \right| \leq \alpha.$$

Without loss of generality, assume that b_1 corresponds to a_1 . Let a'_1 a nearest neighbor of a_1 and b'_1 a nearest neighbor of b_1 . By the above argument,

$$\left| \frac{d(a_1, a'_1) - d(b_1, b'_1)}{d(a_1, a'_1)} \right| \leq \alpha.$$

Thus the theorem follows immediately by applying Theorem I.

5.2 Example Using SNN Algorithm

In this section, an example using the sorted nearest neighbor vector is discussed. Fig 5-1(a), (b) and (c) show three different ideal patterns (even though they "look" similar). Fig 5-2(a)-(c) show the same three patterns but with the nearest neighbors connected.

Let the distances to the nearest neighbor in each pattern be sorted and then calculate the range allowed for the nearest neighbor distance using the results of Thm II with $\alpha = 10\%$. The list of sorted nearest neighbors with the allowed ranges is shown in Fig 5-3.

Now suppose the noisy pattern that is to be classified has the sorted nearest neighbor distances NP(1) thru NP(5).

Using Thm II, these distances must satisfy the following inequalities in order to be a noisy version of pattern A.

$$\begin{aligned}4.5 &\leq \text{NP}(1) \leq 5.5 \\9.0 &\leq \text{NP}(2) \leq 11.0 \\13.5 &\leq \text{NP}(3) \leq 16.5 \\18.0 &\leq \text{NP}(4) \leq 22.0 \\22.5 &\leq \text{NP}(5) \leq 27.5\end{aligned}$$

Similar sets of inequalities must be satisfied in order for a noisy pattern to be classified as noisy versions of patterns B and C.

If the sorted interpoint distances of the noisy pattern are 5.7, 9.7, 13.8, 19.2 and 24.0, then it is a noisy version of pattern B.

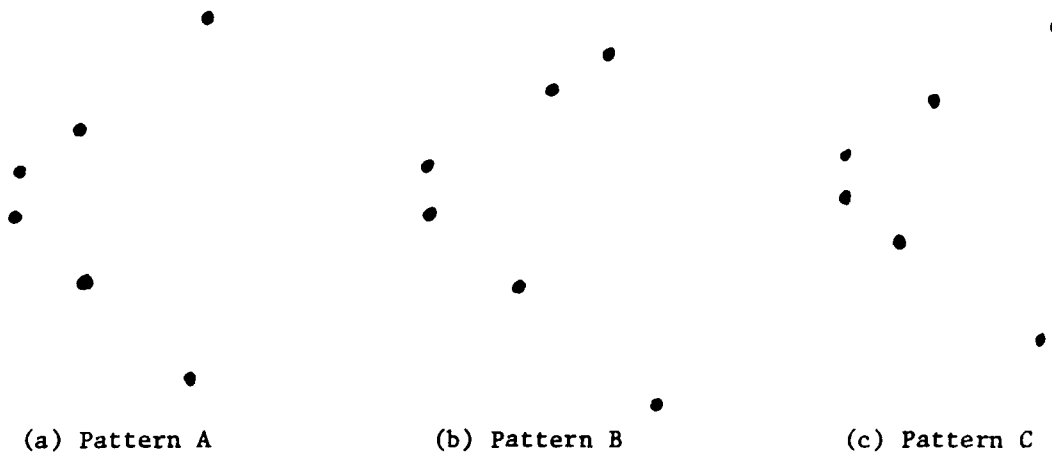


Fig 5-1. Three different ideal patterns used to illustrate SNN classification.

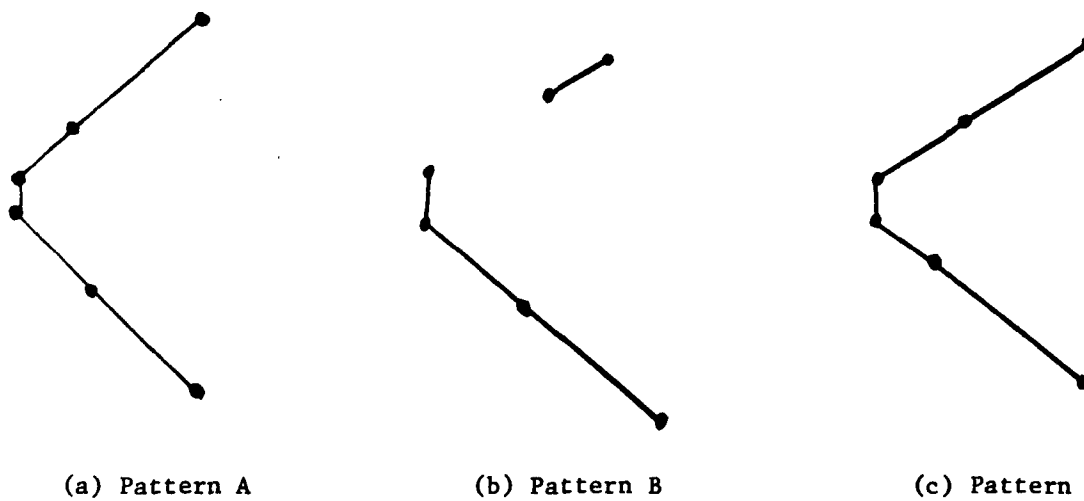


Fig 5-2. The three patterns of Fig 5-1 are shown with the nearest neighbors connected.

<u>Pattern A</u>	<u>Pattern B</u>	<u>Pattern C</u>
5 (4.5-5.5)	6.2 (5.6-6.8)	6.5 (5.8-7.2)
10 (9.0-11.0)	10 (9.0-11.0)	10 (9.0-11.0)
15 (13.5-16.5)	14 (12.6-15.4)	17.5 (15.5-18.9)
20 (18.0-22.0)	20 (18.0-22.0)	20 (18.0-22.0)
25 (22.5-27.5)	25 (22.5-27.5)	25 (22.5-27.5)

Fig 5-3. Sorted nearest neighbor distances for the three patterns plus the allowed range of nearest neighbor distances, in parentheses, assuming 10% noise.

5.3 Experimentation Using SNN Algorithm

In order to test the SNN algorithm, nineteen ideal patterns with twenty points each were generated. Each roughly covered the same area of 100 by 100 units. The twenty patterns are shown in Appendix II.

The patterns were completely distinguishable using an $\alpha = 10\%$. The classification algorithm is shown in Fig 5-4. Thus any α -noisy version of the nineteen patterns would be perfectly classified.

SNN Classification Algorithm for 19 Synthetic Patterns

- (1) If (NP(14)>8.2) then NP is pattern A.
- (2) If (NP(12)>8.2 and NP(20)>15.) then NP is pattern B.
- (3) If (NP(5)>8.2 and NP(20)>18.) then NP is pattern C.
- (4) If (NP(10)>7.1 and NP(20)>18.) then NP is pattern D.
- (5) If (NP(16)>11.7 and NP(9)>8. and NP(20)>30.) then NP is pattern E.
- (6) If (NP(20)>21. and NP(17)>16.8 and NP(7)>8.4 and NP(8)>8.4 and NP(16)>12.1)
then MP is pattern F.
- (7) If (NP(9)>7. and NP(20)>25.) then NP is pattern G.
- (8) If (NP(20)>30. and $8.2 < NP(1) < 11.8$ and NP(4)>11.8 and NP(10)>11.2)
then NP is pattern H.
- (9) If (NP(20)>40. and NP(20)>35. and NP(7)>8.2 and NP(19)>25.) then NP is
pattern I.
- (10) If (NP(20)>40. and NP(8)>8.2) then NP is pattern J.
- (11) If (NP(20)>34. and NP(19)>22.) then NP is pattern K.
- (12) If (NP(19)>26. and NP(1)>8.2) then NP is pattern L.
- (13) If (NP(1)>8.2 and NP(5)>8.2) then NP is pattern M.
- (14) If (NP(1)>8.2 and NP(5)>8.2 and NP(6)>8.2) then NP is pattern N.
- (15) If (NP(1)>8.2 and NP(5)>8.2 and NP(6)>8.2 and NP(12)>11.8) then NP is
pattern O.
- (16) If (NP(1)>8.2 and NP(5)>8.2 and NP(6)>8.2 and NP(12)>11.8) then NP is
pattern P.
- (17) If (NP(1)>11.8) then NP is pattern Q.
- (18) If (NP(4)>11.8) then NP is pattern R.
- (19) Otherwise NP is pattern S.

Fig 5-4. The SNN algorithm for classifying a noisy pattern as one of the nineteen synthetic patterns.

6. Minimal Spanning Tree Methods

The SIDV and SNN do not explicitly encode any global information about a pattern. The only information used in constructing an SIDV or SNN is the distance between two points and includes no information about the structure. Nevertheless experiment indicates the potential of these vectors for classification. In order to explicitly capture more global pattern information, the minimal spanning tree of a pattern P , denoted $MST(P)$ was explored. A series of papers [5,6,7] developed applications of the MST to clustering, bubble chamber pictures, and noisy template matching. Several results, which are useful in analyzing the MST methods, are given in this section.

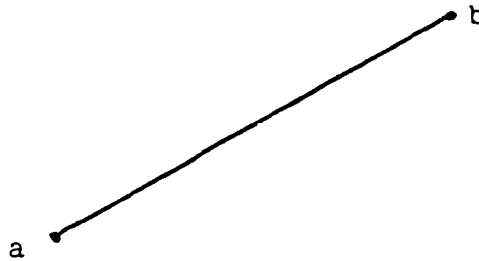
6.1 Graph Theoretical Definitions of MST's

Basic notions from graph theory [2] will be presented in this section. Applications to point pattern classification will be given in the next section. Graph theory is the study of structures composed of points and lines connecting them. A graph is a finite set of points in R^n and a set of lines connecting some pairs of points.* For present purposes our graphs will lie in R^2 . The points in the graph are called vertices and the lines are called edges.

* R denotes a space of real numbers and n denotes the dimension of the space.

Ex. 6.1

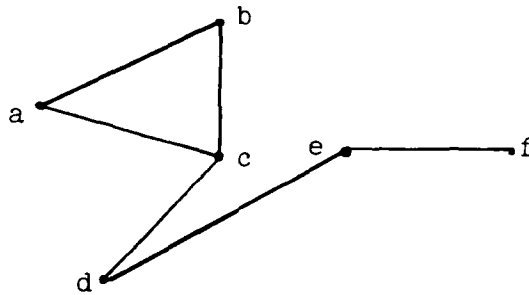
a)



vertices = $\{a, b\}$

edges = $\{(a, b)\}$ where e_{ab} denotes the edges joining vertices a and b.

b)

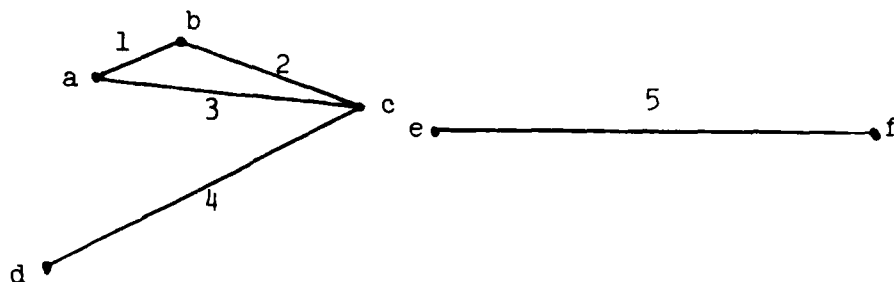


vertices = $\{a, b, c, d, e, f\}$

edges = $\{e_{ab}, e_{bc}, e_{cd}, e_{de}, e_{ef}\}$

The set of vertices of G is denoted $V(G)$ and the set of edges of G is denoted $E(G)$. A weighted graph is a graph, together with an assignment of a positive real number to each edge. The weight of a weighted graph is the sum of the numbers assigned to the edges of G .

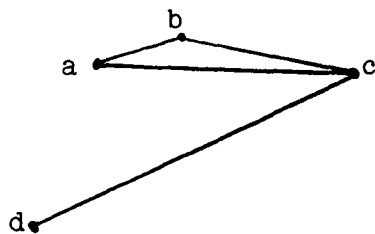
Ex. 6.2



The depicted graph is a weighted graph with the weights written next to the corresponding edges. The total weight of the graph is $1+2+3+4+5 = 15$.

A path in a graph G is a sequence of vertices and edges of G , $v_0 e_1 v_1 e_2 \dots e_k v_k$ where the e_i , $(1 \leq i \leq k)$ are edges, the v_i , $(0 \leq i \leq k)$ are vertices, edge e_i has endpoints v_i and v_{i-1} for $e_i \neq e_j$ for all i and j and no two vertices are the same except for possibly v_0 and v_k . A cycle is a path in which $v_0 = v_k$.

Ex. 6.3



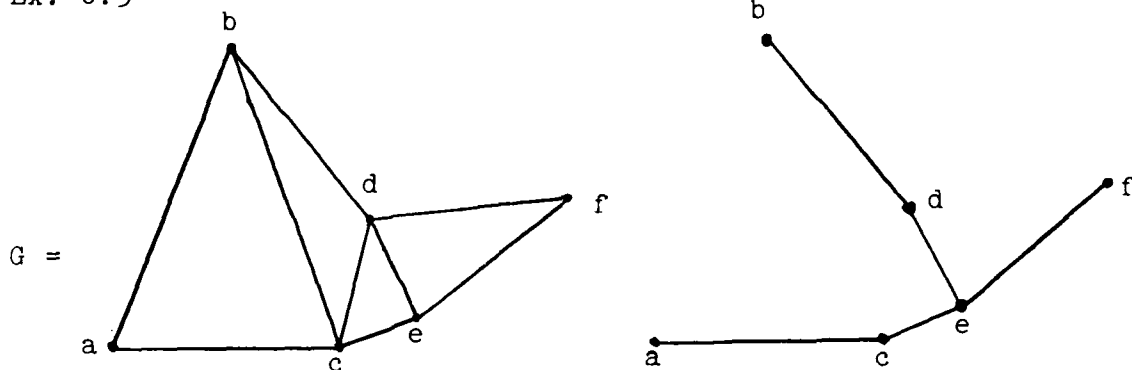
$a, e_{ab}, b, e_{bc}, c, e_{cd}, d$ is a path. $a, e_{ab}, b, e_{bc}, c, e_{ca}, a$ is a cycle.

A graph is called connected if given any two distinct vertices a and b of the graph there is a path $v_0 e_1 v_1 \dots e_k v_k$ such that $v_0 = a$ and $v_k = b$. A tree is a connected graph with no cycles.

Ex. 6.4 The graph of example 6.3 is connected while that of example 6.2 is not.

A spanning tree T of a graph G is a tree whose vertices are exactly those of G and whose edges are a subset (possibly all) of the edges of G .

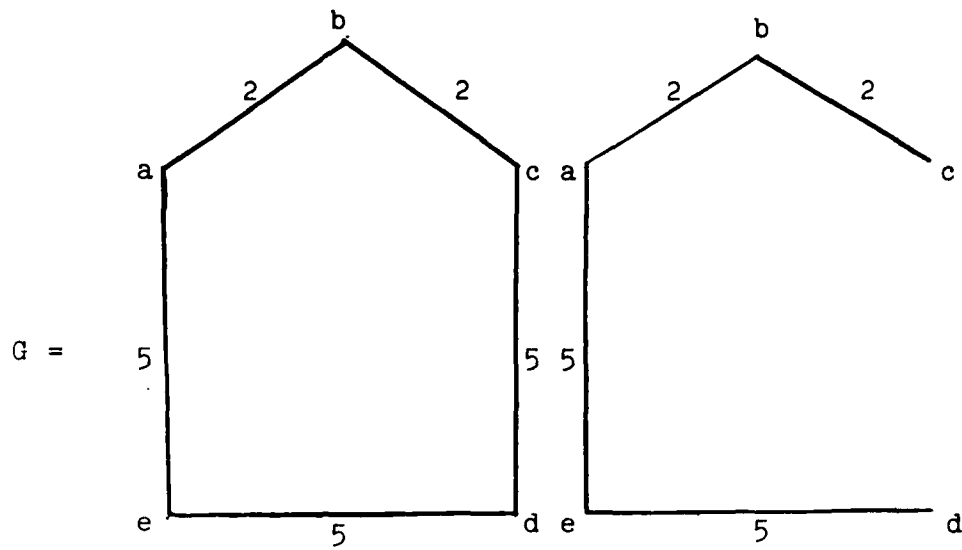
Ex. 6.5



A graph G and a spanning tree T , of G are shown above.

A minimal spanning tree of a weighted graph is a spanning tree of G such that no other spanning tree of G has less weight.

Ex. 6.6



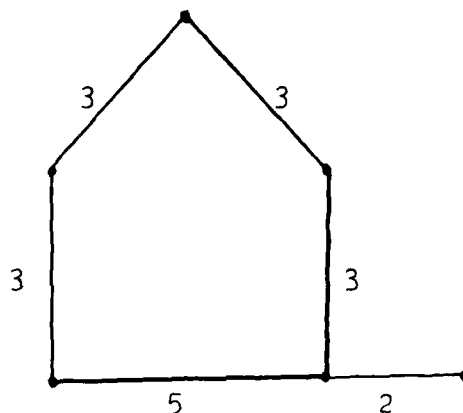
A weighted graph G and a MST T for G are shown above.

6.2 Theory of Using MST's for Classification

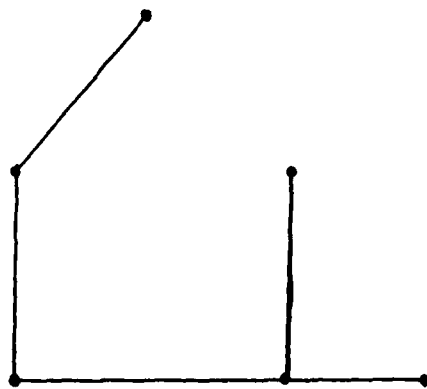
The SIDV of a pattern is uniquely determined by the pattern and is thus independent of the order in which the points in the pattern are processed by the algorithms. This is, unfortunately, not true for the MST. Any graph in which some point has a non-unique nearest neighbor has at least two MST's, which need not be isomorphic.

Ex. 6.7

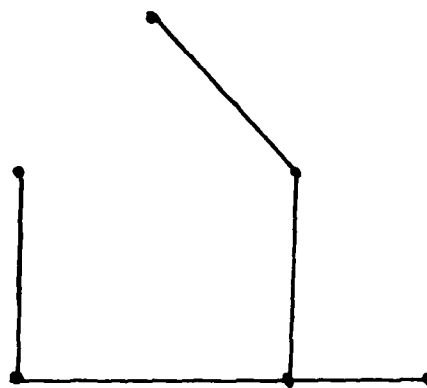
$G =$



$T =$



$T^1 =$



T and T^1 are distinct MST's for the graph G .

Lemma II Let P be a pattern. Let G be the complete graph on P and let S be a set of edges in G that are known to lie in every MST for G . Let G_S be the subgraph of G consisting of the edges in S . Let $C_S = C_1, \dots, C_w$ denote the set of connected components of G_S . Define the distance, $d(C_i, C_j)$, between two components C_i and C_j by

$$d(C_i, C_j) = \min_{e \in E(G)} L(e)$$

where e has one end in C_i and the other in C_j . Let $f \in E(G) - S$. Assume there exists an i, j such that

$$d(C_i, C_j) = L(f)$$

and f has one end in C_i and one end in C_j and

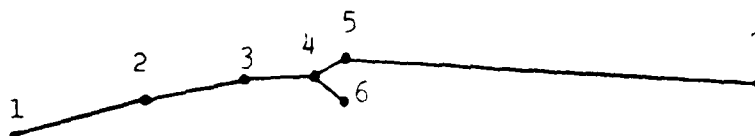
$$d(C_i, C_j) \leq d(C_i, C_k)$$

for $k \neq i, j$ and for all $g \in E(G)$ such that g has one end in C_i and the other in C_j , we have $g \neq f$ implies $L(f) < L(g)$. Then f is in every MST of G .

The above result provides information on the possible variations among MST's for a given point set. Having obtained some understanding of this variability, we now turn to the variability in MST's for an α -percent perturbation of a point pattern. If one can determine a set of edges, E , contained in every MST of a point pattern P , such that any α -percent perturbation P' of P contains a subgraph consisting of an α -percent perturbation of the edges of

E, then we have a potentially powerful tool for matching patterns.

Ex. 6.8



Pattern P

The edges e_{12} , e_{23} , e_{34} are in any MST of pattern P by Thm III. Furthermore as we will show in Thm IV, any 10-percent perturbation P' of P will have the following property. Given any MST M of P' , there exists 3 edges e_{ab} , e_{bc} , and e_{cd} where a, b, c , and d are vertices of M , no two of which are equal and

$$\frac{|(e_{ab}) - (e_{12})|}{(e_{12})} \leq 0.1$$

$$\frac{|(e_{bc}) - (e_{23})|}{(e_{23})} \leq 0.1$$

$$\frac{|(e_{cd}) - (e_{34})|}{(e_{34})} \leq 0.1$$

Thus given a pattern Q , one test that can be applied to determine if it could be an 10-percent perturbation of P is to compute an MST for Q and determine if M has 3 adjacent edges satisfying the above inequalities.

If such equal edge lengths are the only sources of non-uniqueness in MST's then one might determine the extent to which this non-uniqueness hinders matching. We now show that the nearest neighbor equal edges lengths are the only sources of non-uniqueness in MST

Thm III Let P be a pattern, Let G be the complete graph on the pattern and let M be a minimal spanning tree of G . If e_{vw} is an edge of G such that either

- 1) $d(v,w) \leq d(v,z)$ for $z \in V(G)$, $z \neq v$, $z \neq w$
- or
- 2) $d(v,w) \leq d(w,z)$ for $z \in V(G)$, $z \neq v$, $z \neq w$.

Then $e_{vw} \in E(M)$.

Proof: Assume $e_{vw} \notin M$. Then $M \cup \{e_{vw}\}$ is a graph with exactly one cycle and this cycle contains the edge e_{vw} .^{*} The cycle must contain edges f, g $f \neq g$ such that f is incident on w and $f \neq e_{vw}$, or $g \neq e_{vw}$. If such edges did not exist, M would not be connected. $G' = M \cup \{e_{vw}\} - \{f\}$ is a connected graph since removing an edge from a cycle can not disconnect it. Furthermore $L(e_{vw}) < L(f)$ where L denotes the weight of the edge. Thus $L(G') < L(M)$, contradicting the minimality of M . Hence $e_{vw} \in M$.

We can show that this procedure for deciding which edges can be in an MST can be iterated. As a corollary of this result, we will see that if no two interpoint distances in a point pattern are the same, the MST is uniquely determined [5].

^{*} U denotes the operation of set Union.

Thm IV Let P be a pattern, let G be the complete graph on P .

Let α be a real number greater than zero and let P' be an α -percent perturbation of P . Let G' be the complete graph on P' and let M be a MST for G' . If e_{vw} is an edge of G such that either

$$1) d(v,w) \leq d(v,z) \quad \text{for } z \in V(G), z \neq v, z \neq w$$

or

$$2) d(v,w) \leq d(w,z) \quad \text{for } z \in V(G), z \neq v, z \neq w.$$

Then $e_{vw} \in E(M)$.

This follows immediately from Theorems I and III.

The MST of a point pattern can be used for many types of pattern matching. We have experimented with three procedures 1) matching of degree sequences, 2) matching of longest linear paths and 3) matching of adjacent pairs of edges. The first and third procedures depend only upon the measurement of the edges incident upon a point. Thus these procedures are in the spirit of the SIDV or the SNN in that they attempt to match whole patterns by requiring a set of local measurements to match up. In contrast, to the earlier procedures, the measurement generally takes into account at least two edges simultaneously, thus providing a slightly more global set of local operations. Modifications of the adjacent edges permit one to take into account far more global structure into the local operations but at greater computational expense.

6.3 Examples of MST's

Minimal spanning trees were generated for the nineteen ideal patterns used for experimentation in the nearest neighbor algorithm. They are shown in Appendix III. One noisy version of each ideal pattern was calculated and their MST's are shown in Appendix IV.

It is easy to see that the MST does change under noise but that the global structure seems to remain stable.

6.4 Experimentation with MST's

Experimentation of classification using the MST algorithm is incomplete at the end of this phase of the investigation. Attempts to classify using matching of degree sequences or matching using longest linear paths did not work on the patterns discussed above.

A degree of a vertex (point) in a MST is the number of edges coming out of that node. A degree sequence of a MST is then the sorted list of degrees of the points in the MST. Fig 6-1 shows the degree sequences of the MST's for the ideal patterns and their noisy patterns. There was not enough of a match, in general, between the degree sequences of an ideal pattern and its noisy version to devise a classification scheme.

The next attempt in finding a classification scheme used longest linear paths. Longest linear paths are found by starting at a vertex of degree one, and following edges till reaching a vertex of either degree one, or greater than degree two. The number of edges in the path is its length. Fig 6-2 shows the sorted longest linear paths for the ideal patterns and their noisy patterns. Again, a classification scheme using the longest linear paths was not possible.

	<u>IDEAL</u>						<u>NOISY</u>					
	Degree Sequence						Degree Sequence					
<u>Pattern</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>
A	8	7	4	1			8	7	4	1		
B	7	8	5				7	8	5			
C	5	12	3				6	10	4			
D	5	12	3				5	12	3			
E	4	14	2				4	14	2			
F	6	10	4				7	8	5			
G	4	14	2				6	10	4			
H	7	8	5				7	8	5			
I	4	14	2				6	10	4			
J	6	10	4				6	10	4			
K	5	12	3				4	14	2			
L	7	9	3	1			6	11	2	1		
M	4	14	2				6	10	4			
N	4	14	2				5	12	3			
O	6	10	4				6	11	2	1		
P	7	8	5				7	8	5			
Q	4	14	2				4	14	2			
R	5	13	1	1			6	11	2	1		
S	4	13	3				5	12	3			

Fig 6-1. The degree sequences for the ideal and noisy MST's for the nineteen synthetic patterns is shown. For example, the degree sequence for the ideal MST for pattern A is 11111111 2222222 3333 4.

8 7 4 1
 6-12

Patterns	<u>IDEAL</u> Length of Linear Paths														<u>NOISY</u> Length of Linear Paths										
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11
A	8	2	1	1											8	3			1						
B	6	3	1	1											7	1	2	1							
C	1	3	1	1	1										4	3		1	1						
D	3		1	2	1										4			1	1	1					
E	2	1			1					1					2	1			1						1
F	5	1	1	1	1										6	2	3								
G	2	1	1									1			3	3	2	1							
H	5	4	2												5	4	2								
I	3	1												1	8										1
J	3	4	1		1										3	3	2	1							
K	3	1	1	1			1								1		1	2		1					
L	3	2	1	1	1										3	5	2								
M	4	3		1	1											1		3	1						
N	1	1		2				1							4		1	1			1				
O	4	2	1	2											3	1	2	2							
P	6	3	1	1											6	3	1	1							
Q	2		1	1						1					1	2		1						1	
R	2	2	1		1		1								4	1		2	1						
S	1	6	2		1		1								2	2	1	1	1	1					

Fig 6-2. The length of linear paths for the ideal and noisy MST's for the nineteen synthetic patterns is shown. For example, the length of linear paths for the ideal MST for pattern a is 11111111 22 3 4.

8 2 1 1

The final attempt was to match MST's by their adjacent edge lengths. For this technique all possible adjacent edge lengths of the tree are found and sorted. Then classification is done by deciding whether the sequences from two trees match.

The comparison of adjacent edge lengths is much more complicated than the comparison of distances done in the SIDV and SNN algorithms. In SIDV and SNN algorithms each component of the sequence was composed of only single numbers and there was an obvious way to sort and compare them. Here, in the adjacent edge length case, each component is composed of two numbers and the method of sorting and comparing is not as obvious.

Testing of this method is not complete, but preliminary results show that this method might give a good classification scheme.

7. Patterns Specified by a Set of Constraints

The point patterns considered in sections 3-6 were perturbations of some ideal configurations of points. The perturbations considered were subject only to the restriction that no interpoint distance should change by more than a fixed percentage. We now describe an important class of point patterns, which we call constrained patterns which include the α -percent perturbation model. Unlike the α -percent perturbations models, the constrained patterns do not admit a simple class of general recognition algorithms, though we present an example to illustrate how certain realistic instances can be handled quite easily.

A constrained pattern is merely a set of conditions which can be tested on a point set in the plane. If a point set satisfies all the conditions it is called an instance of the pattern. Different classes of constrained patterns can be defined depending on the nature of the conditions. Examples of useful conditions include restrictions on interpoint distances, angles formed by lines joining certain points, total size of the pattern and minimum distance between certain points. Little can be said about the problem at this level of generality.

7.1 An Example

We now give an example of a pattern closely related to an existing real system. The pattern, shown in Fig 7-1, consists of seven points. Three points form the vertex of an equilateral triangle of side length 50 feet at the vertex of the V. None of the remaining four points can be closer than 150 feet from one another or to any of the three points at the vertex. It must be possible to superimpose the shaded V of Fig 7-2 on the pattern in such a way that the mean of the three vertex points lies at the vertex of the V, and regions A and B contain two points each. An algorithm that gives a complete solution to the recognition problem for this pattern is presented below. Any set of points satisfying the constraints of the pattern will be labeled by the algorithm as an instance of the pattern. We first give an informal statement of the algorithm which we call Algorithm A.

Algorithm V (Informal)

- 1) Locate the vertex of the V by finding three points, a, b, c, whose interpoint distances are all 50 feet. Find the mean m of these three points.
- 2) Project all points other than a, b, and c onto the unit circle with center m.

Compute the angles v_1, \dots, v_4 formed by the consecutive points on the circle. If exactly one of the v_i is greater than 150° , then state this is not a V and terminate.

- 3) Renumber the subscripts of the v_i 's cyclicly if necessary so that $v_4 - v_1 \leq 150^\circ$.
- 4) Test the v_i 's for the satisfaction of inequalities required by the V pattern (to be discussed next). If all the inequalities are satisfied, an instance of the pattern has been found, else the points do not form an instance of the pattern.

We now give a formal definition of the V pattern.

Def. A V pattern is a set $P = \{a_1, \dots, a_7\}$ of seven points in R^2 where the coordinates of the point a_i are (x_i, y_i) . The set P must satisfy the following conditions:

- 1) $d(a_5, a_6) = d(a_5, a_7) = d(a_6, a_7) = 50$ where d denotes Euclidean distance

- 2) Let m be the point with coordinates (m_1, m_2) where

$$m_1 = 1/3(x_5 + x_6 + x_7)$$

$$m_2 = 1/3(y_5 + y_6 + y_7)$$

For $i = 1, 2, 3, 4$, $d(a_i, m) \leq 1000$

- 3) For $i = 1, 2, 3, 4$, $j = 1, \dots, 7$ $d(a_i, a_j) \geq 150$
- 4) The following figure can be superimposed on the plane in such a way that points O and m coincide and points a_1 and a_2 lie in one shaded region, while points a_3 and a_4 lie in the other.



C

Fig 7-1. An example of the V pattern.

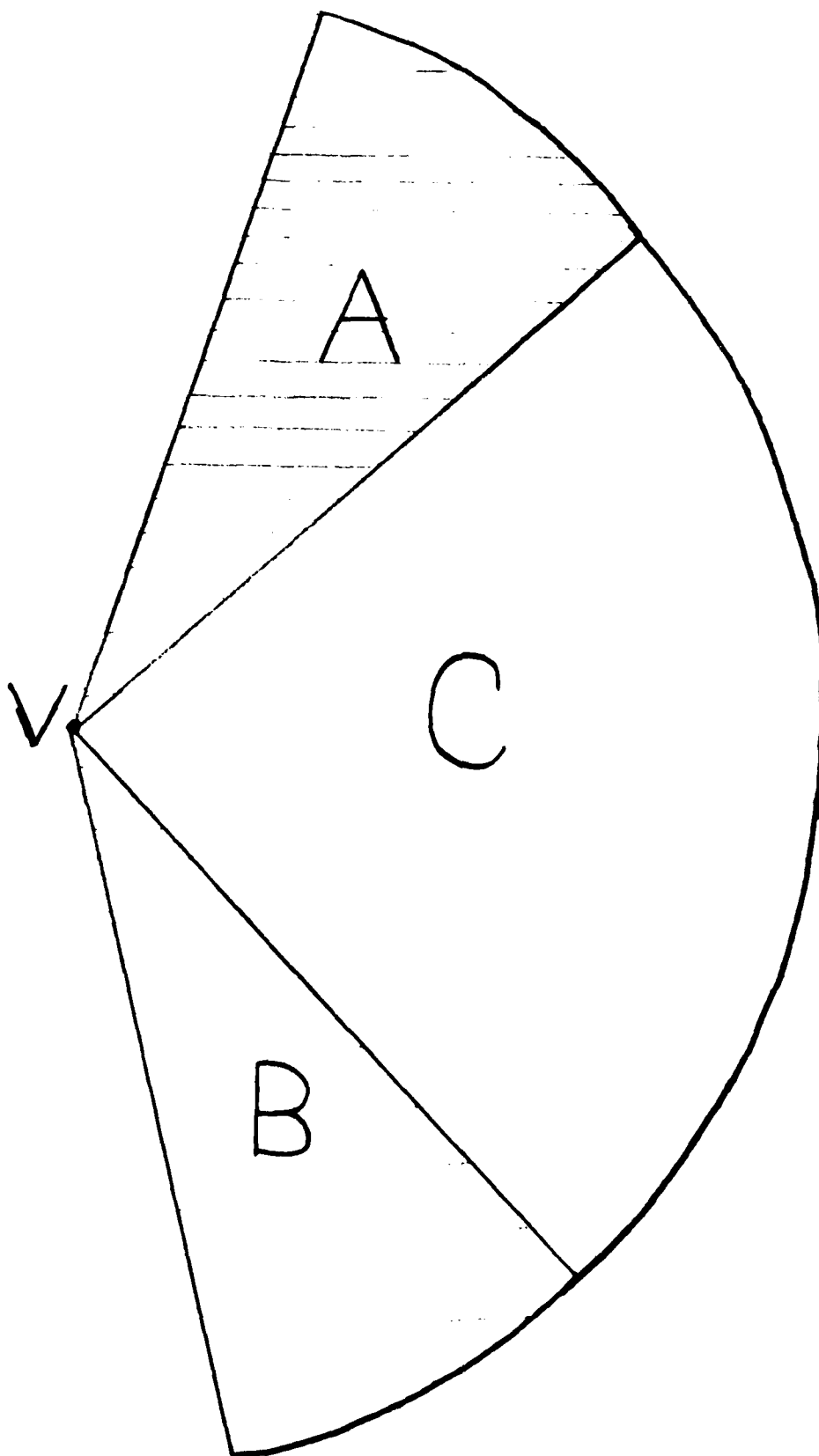
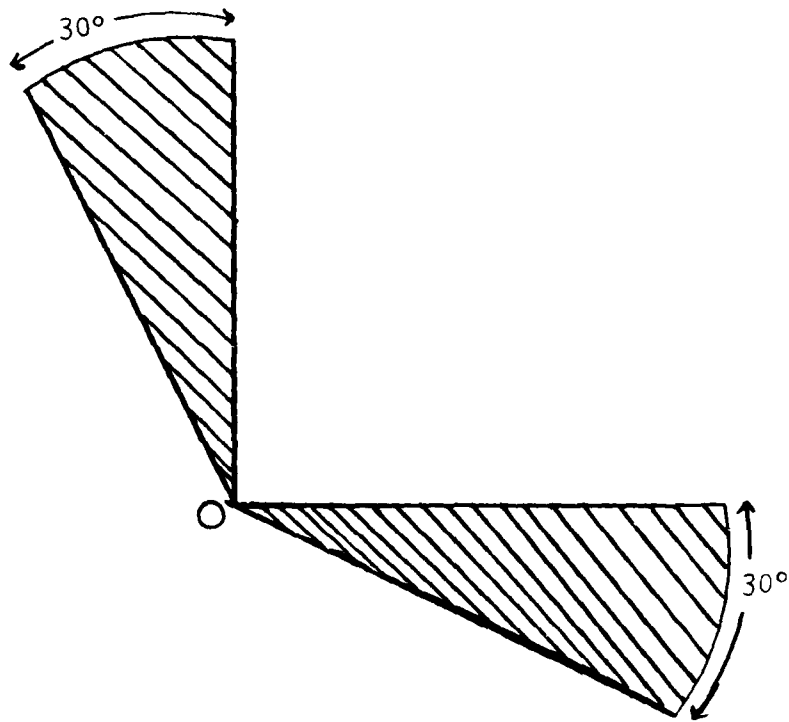


Fig 7-2. The three regions of the V-pattern



The curved segments are circular arcs from a circle of radius 1000 and center O .

We now use the above definition to define a set of inequalities which can be used to test whether the sectors described above can be imposed on a set of points. Assume we have an instance of the V pattern. Relabeling our points if necessary we may assume $A(a_1, a_2)$, $A(a_1, a_3)$, $A(a_1, a_4)$ are less than 150° and $d(m, a_i) < 1000$ feet for $i = 1, \dots, 4$.

Clearly $0^\circ \leq A(a_1, a_2) \leq 30^\circ$

$0^\circ \leq A(a_3, a_4) \leq 30^\circ$

$90^\circ \leq A(a_2, a_3)$

$A(a_1, a_4) \leq 150^\circ$

Conversely we shall show that if these inequalities hold for points a_1, a_2, a_3 and a_4 then we have a V pattern. $A(a_1, a_j)$ denotes the angle formed by a_1 , O , and a_j .

Algorithm for detecting V pattern (Formal):

Basic V pattern - three points at vertex, two on each side of V

- I Compute the interpoint distance matrix $X(I,J)$ $I = 1, \dots, 7$,
 $J = 1, \dots, 7$ where 7 is the number of points in the pattern
and $X(i,j)$ is the distance from point I to point J.
- II In the upper triangular matrix $\{X(I,J)\}$ $I = 1, \dots, N-1$,
 $J = I+1, \dots, N$ compute the number, K, of values $X(I,J)$
satisfying $40 < X(I,J) < 0$. If $K \neq 3$ terminate the algo-
rithm. The pattern is not a V. If $K = 3$ and $X(I_0, J_0)$,
 $X(I_1, J_1)$, and $X(I_2, J_2)$ are the three values found then:
If the set $\{I_0, J_0, I_1, J_1, I_2, J_2\}$ contains exactly
three distinct values go to step 3; else stop. The
pattern is not a V.
- III Let a,b,c, be the three distinct values found in the set
S. Set $m = 1/3(a_1+b_1+c_1, a_2+b_2+c_2)$ where
point a has coordinates (a_1, a_2)
" b " " (b_1, b_2)
" c " " (c_1, c_2)
(M is the vertex of the V)
- IV Let $T = \{Y_1, \dots, Y_4\}$ be the set of points in the pattern
excluding points a, b and c.
Let e_i denote the line segment joining M and Y_i for
 $i = 1, \dots, 4$.
Let f_i denote the angle in the counterclockwise direction
from a horizontal line through to e_i .
If any length (e_i) is greater than 1000 terminate with
failure.

V Sort the values f_i , $i = 1, \dots, 4$ in increasing order.

Denote the sorted angles by v_1, \dots, v_4 .

Compute the values $\{v_{i+1} - v_i\}_{i=1}^3$ and $v_1 - v_4$. Denote this set by Q .

If exactly one element of Q is greater than 150° go to step VI; else terminate with failure.

VI Renumbering the subscripts of the v if necessary, we may assume, without loss of generality, that $v_1 - v_4$ is the one value greater than 150° .

VII Assume $N = 7$

If $v_2 - v_1 \leq 30^\circ$

and $v_3 - v_2 \geq 90^\circ$

and $v_4 - v_3 \leq 30^\circ$

and $v_4 - v_1 \leq 150^\circ$

then terminate with success.

Otherwise terminate with failure.

From the definition of the V pattern, it can easily be shown that any instance of this pattern will be recognized by the algorithm. We now show that the algorithm claims detection for a restricted V pattern only if it truly is a V pattern. To see this, it suffices to show that a V can be superimposed on the pattern if the restrictions on angles given in step VII of algorithm V are satisfied. Thus we assume we have a point in which will be the vertex of our V and four points a_1, a_2, a_3 and a_4 .

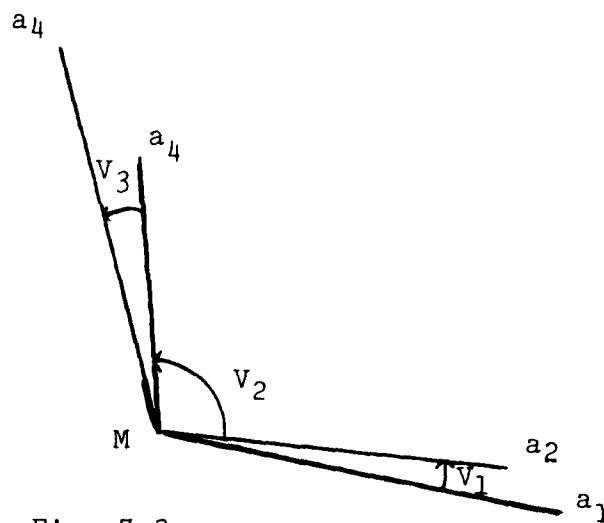


Fig. 7-3.

Let V_1, V_2 , and V_3 be the angles shown in Fig 7-3. We assume $0^\circ \leq V_1 \leq 30^\circ$, $0^\circ \leq V_3 \leq 30^\circ$, $V_2 > 90^\circ$ and $V_1 + V_2 + V_3 \leq 150^\circ$. Let \bar{V} denote the shaded region in Fig 7-4.

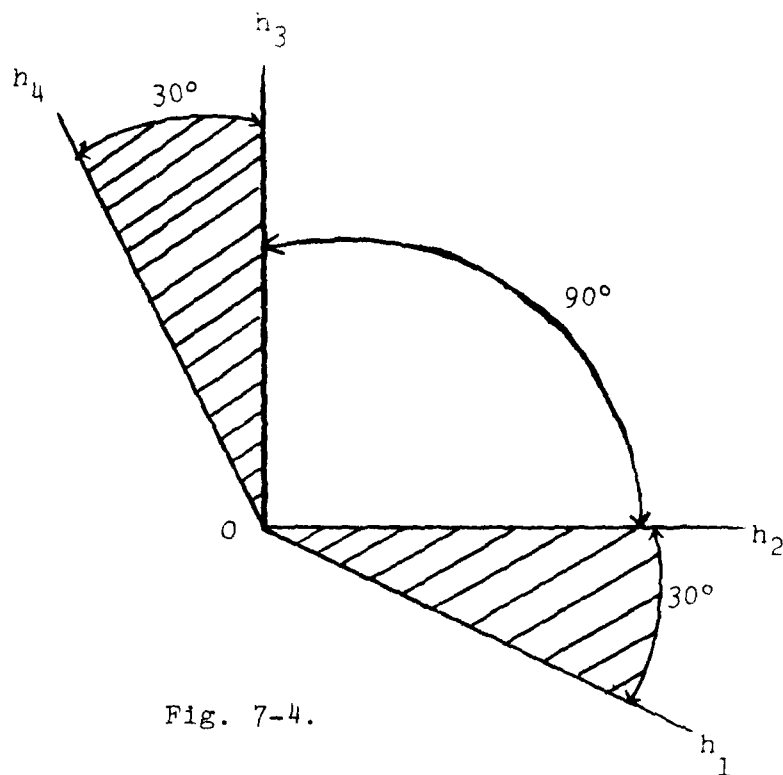
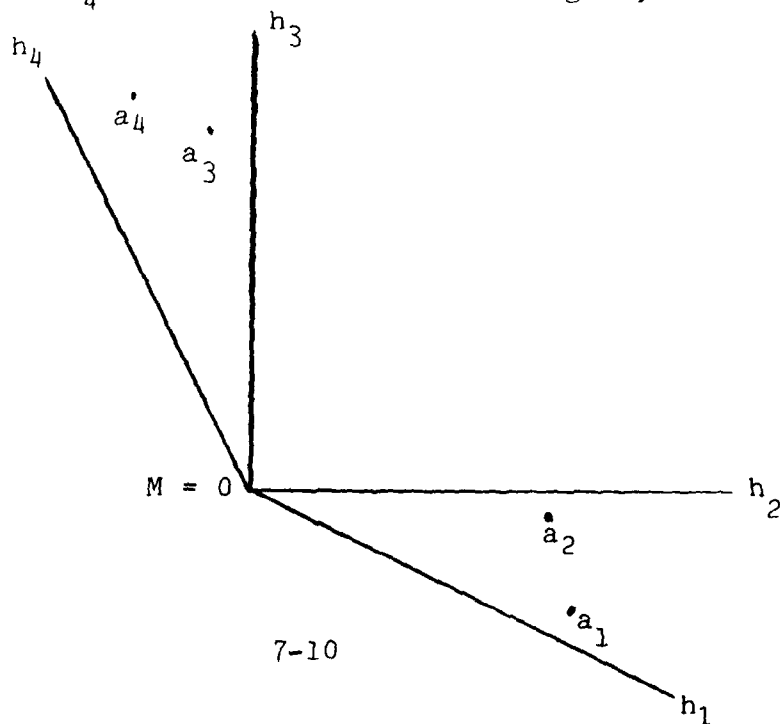
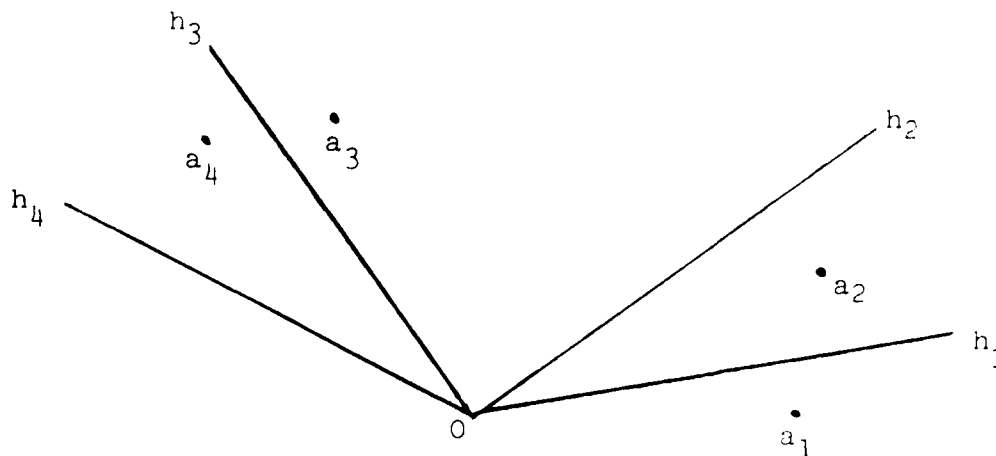


Fig. 7-4.

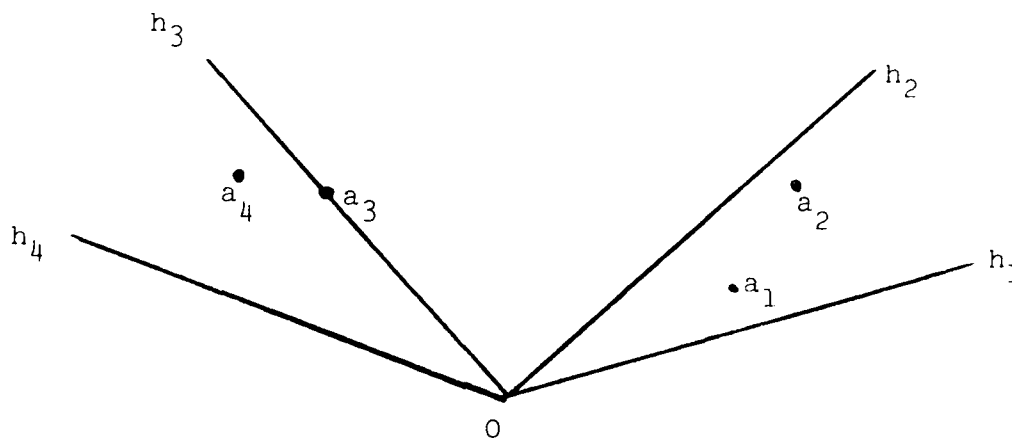
Suppose we superimpose Fig 7-3 on Fig 7-4 by placing point O on point M and line Oh_1 on Ma_1 . Since $V_1 + V_2 + V_3 \leq 150^\circ$, none of the points a_1, a_2, a_3 and a_4 can lie outside the angle h_1Oh_2 . Since $V_1 \leq 30^\circ$, and $h_1Oh_2 = 30^\circ$, both a_1 and a_2 will lie in a shaded region. If a_3 and a_4 lie in the other shaded region, we are done.



If not, then a_3 must lie in the angle h_2Oh_3 .



In this case we rotate \bar{V} clockwise till a_3 is on the line Oh_3 .



Since $V_3 \leq 30^\circ$ both a_3 and a_4 lie in angle h_3Oh_4 . If a_1 and a_2 lie in angle h_1Oh_2 we are done. Neither a_1 nor a_2 can lie outside angle h_1Oh_4 since we shifted clockwise by less than 30° . Thus the only way a_1 or a_2 can fail to lie in angle h_2Oh_3 is if a_2 lies in angle h_2Oh_3 . But if this happens then V_2 is less than 90° . This is a contradiction since we assumed $V_2 \geq 90^\circ$. Thus we can superimpose a V pattern on the points.

8. Conclusions

Two classes of point patterns were investigated. The first class was the prototype pattern with a noise model. We feel that classification of these patterns can be accomplished using the discussed methods, SIDV, SNN and MST, under certain restrictions. If the number of points in the pattern is small, less than 100, then the SIDV method would be better. If the number of points in the pattern is large, more than 100, then the SNN or MST methods would be more useful.

The other restriction concerns the relative number of additions or deletions of pattern points. If the percentage of additions or deletions is small, for example on the order of 10%, then the SIDV, SNN and MST methods are still feasible. If the percentage is large, then other methods should be used. (See Sec 9.2.)

The techniques used to classify prototype point patterns all gave a measure of the quality of the match. This measure is useful since it can be used to provide a measure of the confidence on the classification of a particular pattern.

The second class of point pattern investigated was a pattern specified by a set of constraints. Due to lack of data, only one example was studied. An algorithm that could recognize any instance of this pattern was successfully developed. Generalization of this technique was impossible without more data. (See Sec 9.3.)

9. Recommendation for Future Investigation

9.1 MST

The experimentation using the adjacent edge lengths of the MST for classification should be finished. Variations of this method which take into account the angles of the MST could also prove very useful.

Classification using the adjacent edge lengths is done by sorting the components, which consist of pairs of edge lengths, and comparing the components. This method could be expanded by including the angle between the adjacent edge lengths. The components, which now consist of a pair of edge lengths and an angle, are sorted and compared componentwise.

9.2 Additions and Deletions of Pattern Points

The methods reported for classification, the SIDV and SNN, were very successful for relatively few additions or deletions of pattern points. These methods are particularly attractive because they are simple and fast to compute. However, as the number of additions or deletions of points become large relative to the number of points in the pattern, then classification using the SIDV or SNN could become computationally expensive. In order to evaluate these methods more thoroughly, precise information about the probability of additions or deletions of points needs to be provided.

If this probability turns out to be a significant proportion, then classification using more statistical type of information should be investigated. For example, one class of statistical properties that is independent of translation and rotation is the

moments of the pattern. The ij^{th} moment of the pattern is defined as $m_{ij} = \sum x^i y^j$. To make the moments independent of the choice of axis, the origin should be moved to the center of gravity of the pattern, defined by $\left(\frac{M_{10}}{N}, \frac{M_{01}}{N}\right)$ where N is the number of points in the pattern. The x-axis is then chosen to be the principal axis which is the line through the center of gravity about which the spread of the pattern points is least. Odd moments, now defined with respect to the new coordinate system, will give a measure of the balance of points between left and right, or up or down. Even moments will give a measure of the spread of pattern points away from the axis.

9.3 General Approach to Pattern Specified By Constraints

Although an algorithm for recognizing a pattern specified by constraints (the V-pattern) was easily devised, it was specific to that pattern. In order to devise more general classification schemes, many more patterns specified by constraints need to be studied. The classification schemes would probably utilize spatial and angular relationships among the points.

9.4 Testing of Classification Schemes

Testing of all the devised classification schemes should be performed on actual data from point patterns in order to make a final determination of their relative usefulness.

BIBLIOGRAPHY

- 1) Dickson, L. E., "Algebraic Invariants," John Wiley and Sons, New York, 1914.
- 2) Harary, F., "Graph Theory," Addison Wesley, Reading, 1969.
- 3) Kahl, D. J., A. Rosenfeld, and A. Danker, "Some Experiments in Point Pattern Matching," IEEE Trans on Systems, Man, and Cybernetics Vol. SMC-10, No. 2, February 1980, pp 105-116.
- 4) Simon, J., A. Checroun, and C. Roche, "A Method of Comparing Two Patterns Independent of Possible Transformations and Small Distortions," Pattern Recognition 4, 1972, pp 73-81.
- 5) Zahn, C. T., "Graph-theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans Computers Vol. C-20, No. 1, January 1971, pp 68-86.
- 6) Zahn, C. T., "Using the Minimum Tree to Recognize Dotted and Dashed Curves," Proc of on Intl. Computing Symp. (1973), A. Gunther et al (Editors), North Holland Publishing Company, 1974.
- 7) Zahn, C. T. "An Algorithm for Noisy Template Matching," Proc IFIP Congress (1974), pp 698-701.

Appendix IFormal Definition of Problem

Def A pattern is a finite set of points in R^2 .

Def Let $S = \{P_1, \dots, P_n\}$ be a set, such that each element, P_i , of S is a finite set of points in the plane. Denote the elements of the set P_i by $\{a_{i1}, \dots, a_{ik(i)}\}$ where $k(i)$ denotes the number of points in P_i . For $1 \leq i \leq n$, $1 \leq j \leq k(i)$, let $a_{ij} = (x_{ij}, y_{ij})$ where x_{ij} and y_{ij} are the cartesian coordinates of the point a_{ij} . The set S will be referred to as the set of ideal patterns and each element P_i of S will be called an ideal pattern.

Def Two patterns A and B are said to be equivalent if there exists a translation, rotation, and reflection of R^2 such that some product of those induces a 1-1 map of the point set A into the point set B .

Def A point pattern classification procedure T_S for a set $S = \{P_1, \dots, P_r\}$ ideal patterns is a procedure realizing a function $f_{T_S}: N \rightarrow 2^{(1, \dots, r)}$ where N is the set of all finite point sets in the plane and $2^{(1, \dots, r)}$ is the set of integers $\{1, \dots, r\}$. Intuitively, given a pattern P in N , $f_{T_S}(P)$ consists of the indices of the ideal patterns which could possibly match P .

Def A point pattern classification procedure T_S , $S = \{P_1, \dots, P_r\}$ is called regular if for each P in N such that P is equivalent to P_i for some i in $\{1, \dots, r\}$, we have $i \in f_{T_S}(P)$. Thus a regular point pattern classification procedure always in-

cludes a pattern P_1 as a possible match for a pattern P if P_1 can be obtained from P by translations, rotations, and reflections.

Def A noise model M for a set $S = \{P_1, \dots, P_r\}$ of ideal patterns is a function $M: S \rightarrow 2^N$ where 2^N is the set of all subsets of the set N of finite point sets in the plane. Thus M assigns to any pattern P_1 in S , a set of patterns. We call any element of this set a noisy version of P_1 .

Def The α -percent noise model M_α for an ideal set of patterns $S = \{P_1, \dots, P_r\}$ and $\alpha \geq 0$ is the noise model for S where for any $1 \leq i \leq r$, $M_\alpha(P_i)$ is defined as follows:

A pattern P is an element of $M_\alpha(P_i)$ iff

- 1) The cardinalities of the two sets P_i and P are the same, i.e. $|P| = |P_i|$.
- 2) If $P_i = \{a_1, \dots, a_m\}$ and $P = \{b_1, \dots, b_m\}$ then there is a 1-1 function $g: \{1, \dots, m\} \rightarrow \{1, \dots, m\}$ such that for all $1 \leq k, n \leq m$

$$\frac{|d(a_k, a_n) - d(b_{g(k)}, b_{g(n)})|}{d(a_k, a_n)} \leq \alpha$$

where $d(a, b)$ denotes the Euclidean distance between points a and b .

Intuitively, M assigns to a pattern P those patterns in S which can be obtained by perturbing the interpoint distances in P by no more than α -percent.

Def An α -percent regular point pattern classification procedure

R for an ideal set of patterns $S = \{P_1, \dots, P_r\}$ is a regular point pattern classification procedure such that if f_{R_S} is the function corresponding to R_S , then $P \in M(P_1)$ implies $1 \in f_{R_S}(P)$ for $1 \leq i \leq r$. Hence such a classification procedure is merely a procedure which designates P_1 as a possible match for P if their interpoint distances match to within α -percent.

Appendix 2

Nineteen ideal patterns of twenty points each used to test
the SNN and MST techniques

Appendix 3

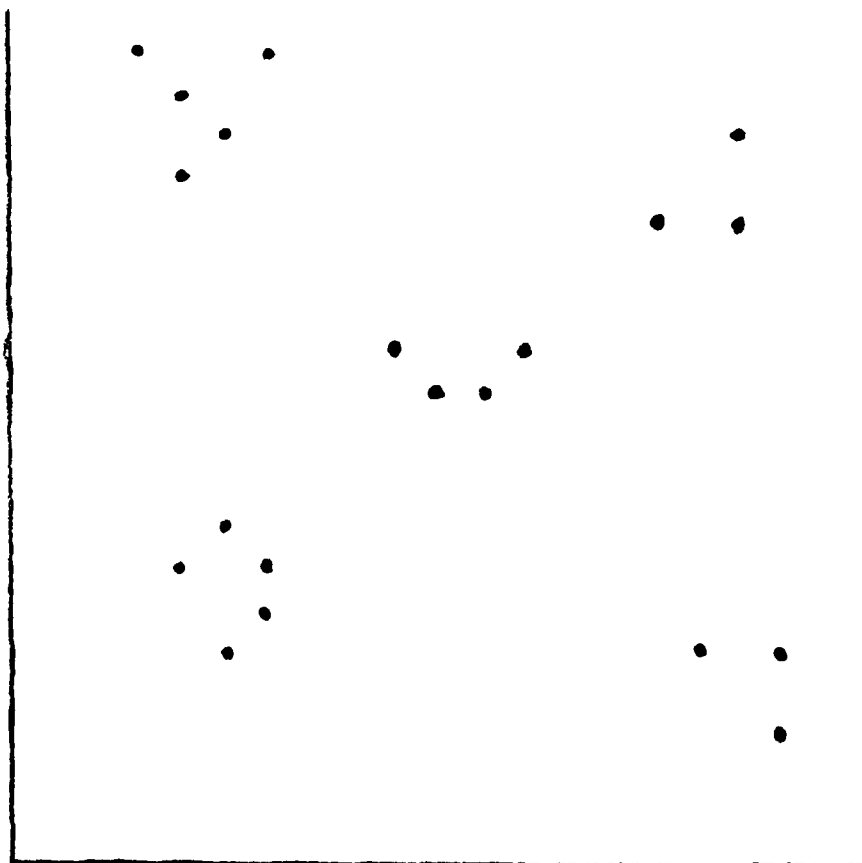
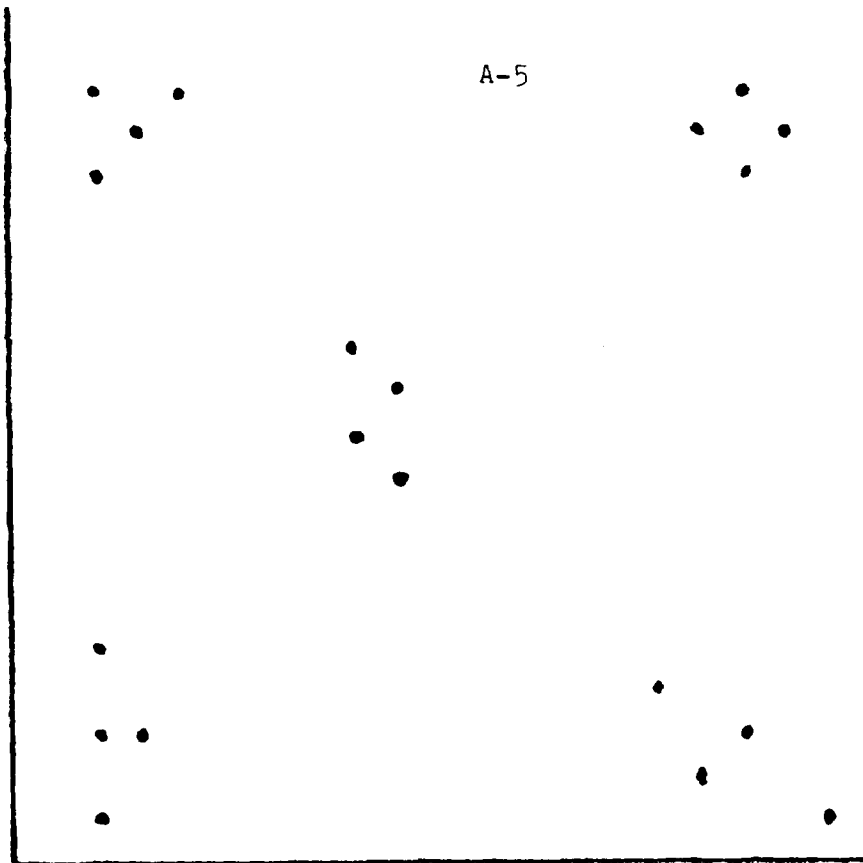
Minimal spanning trees of the ideal patterns in appendix 2

Appendix 4

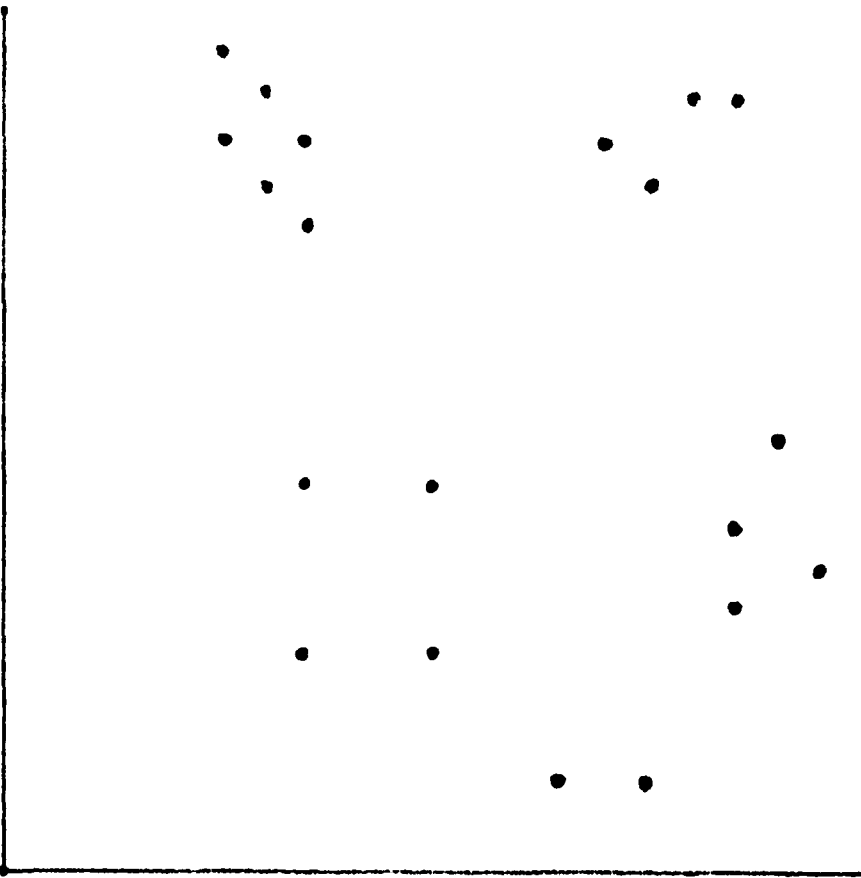
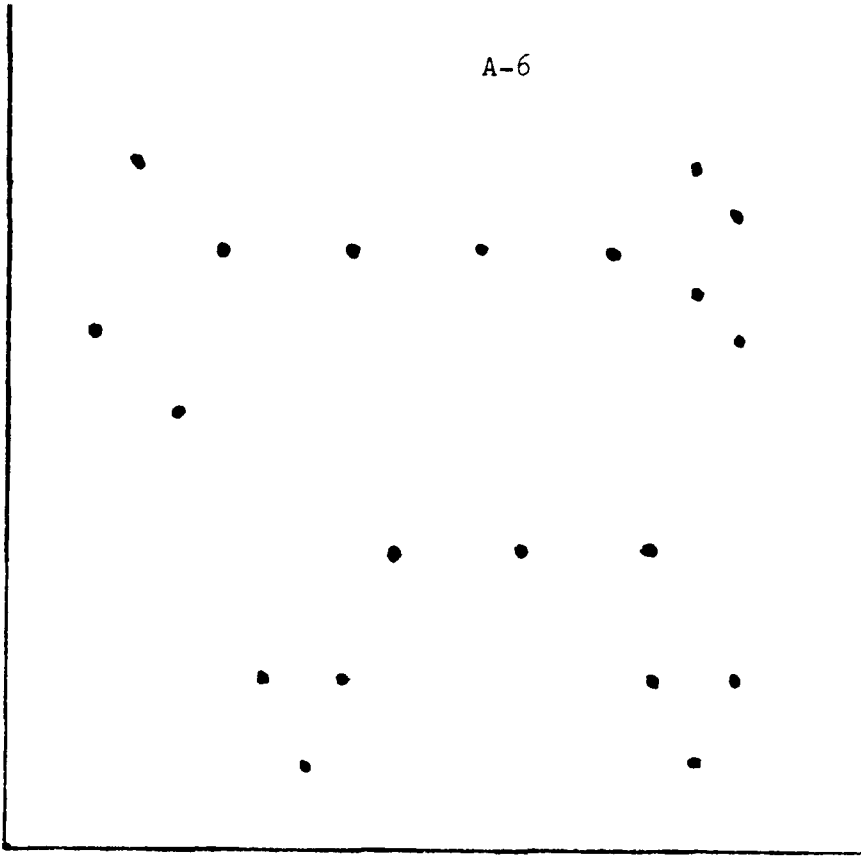
Minimal spanning trees of noisy versions of the patterns in
appendix 2

A-5

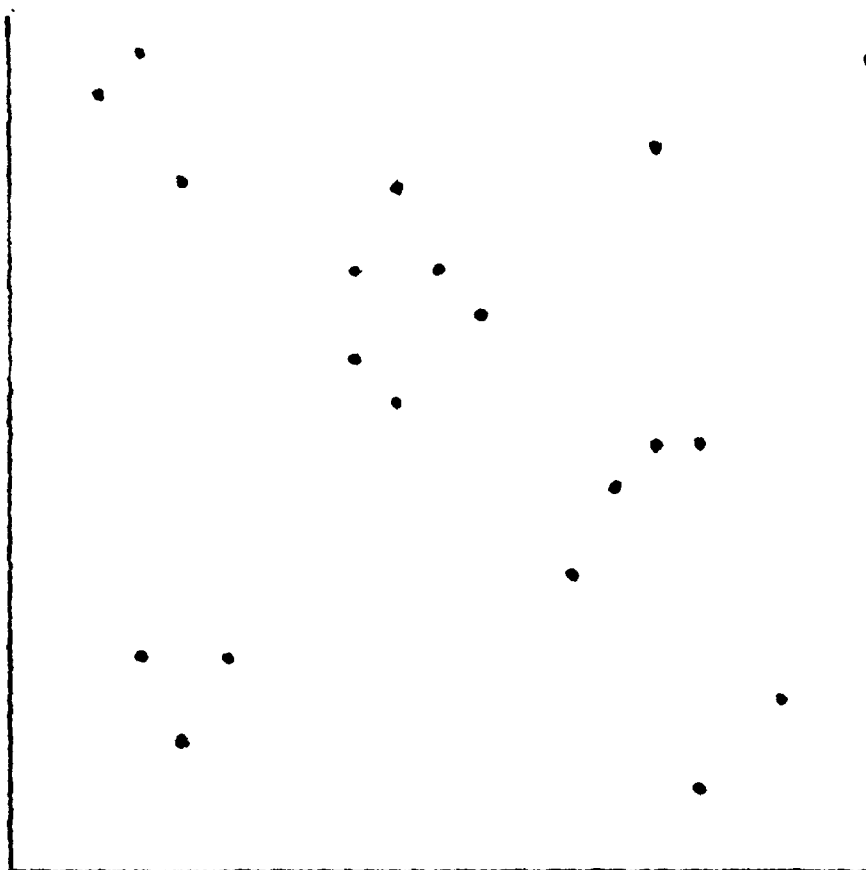
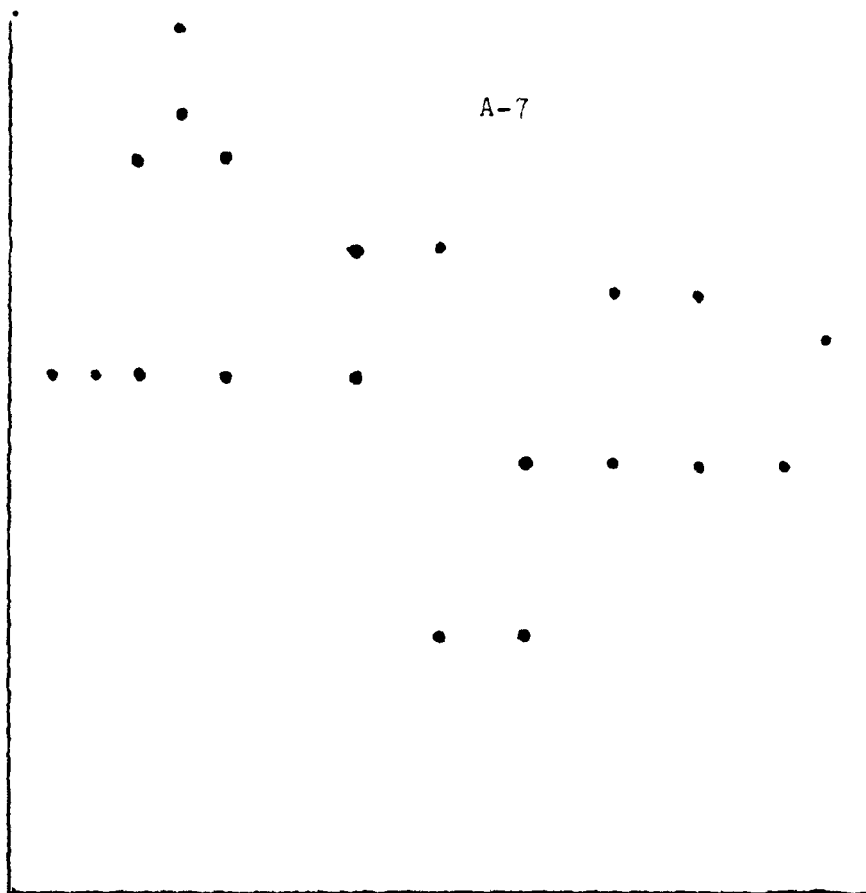
Appendix 2



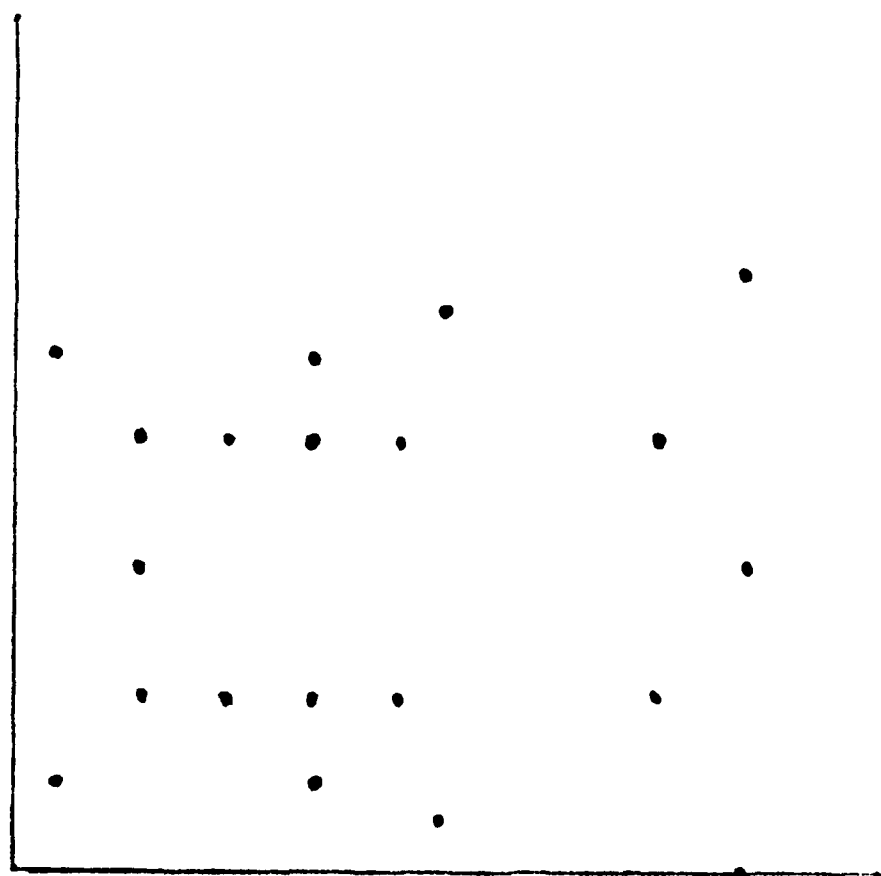
A-6



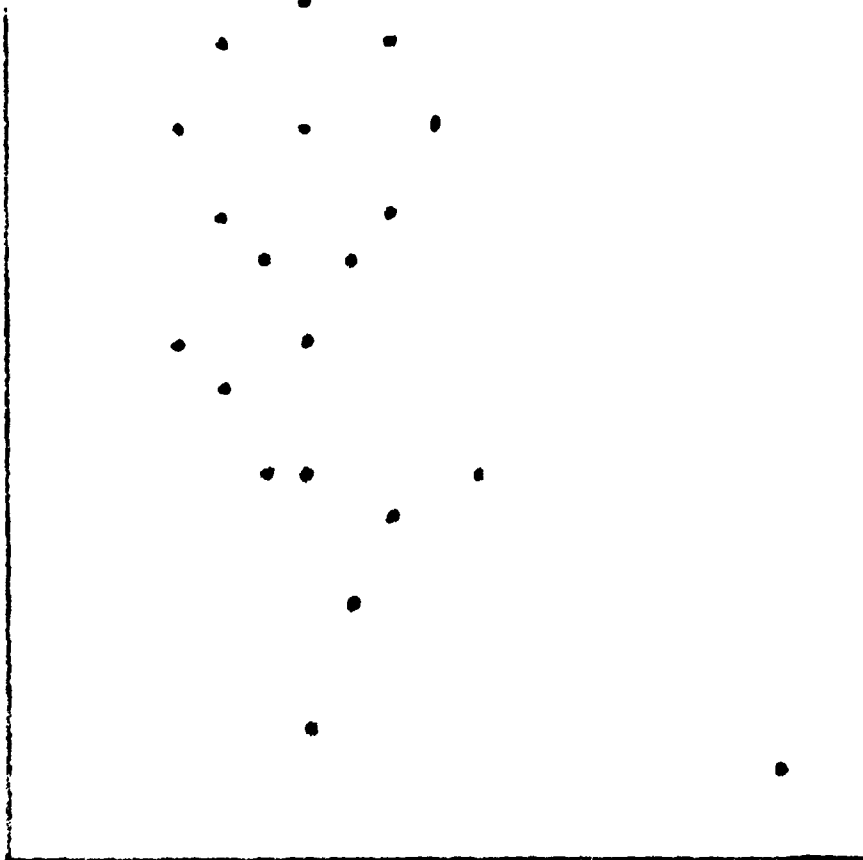
A-7



A-8



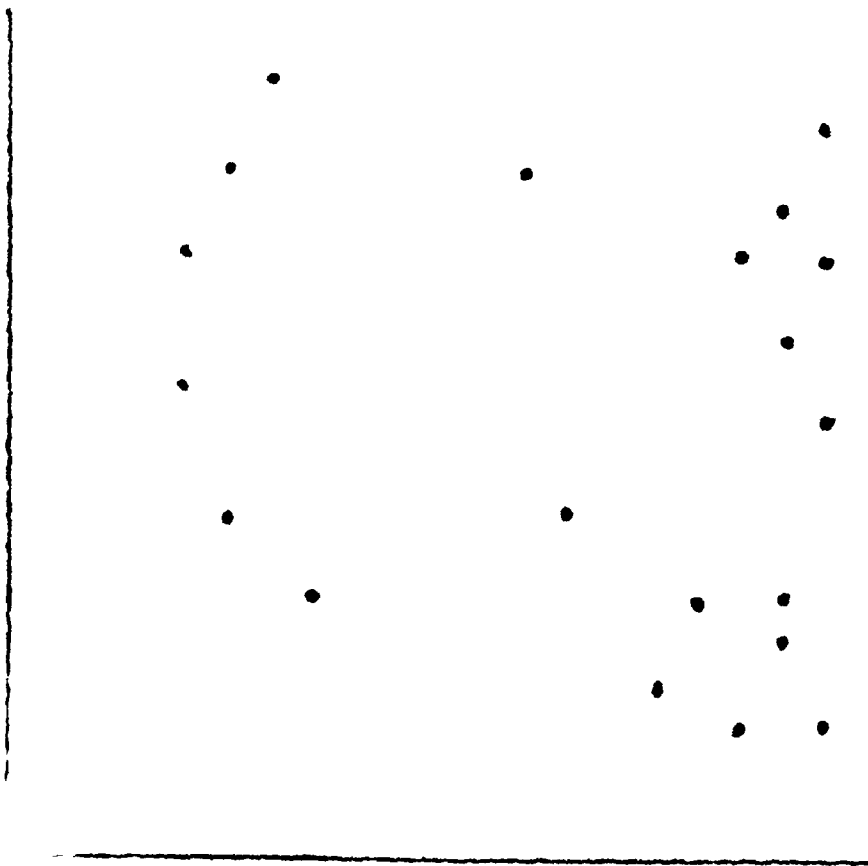
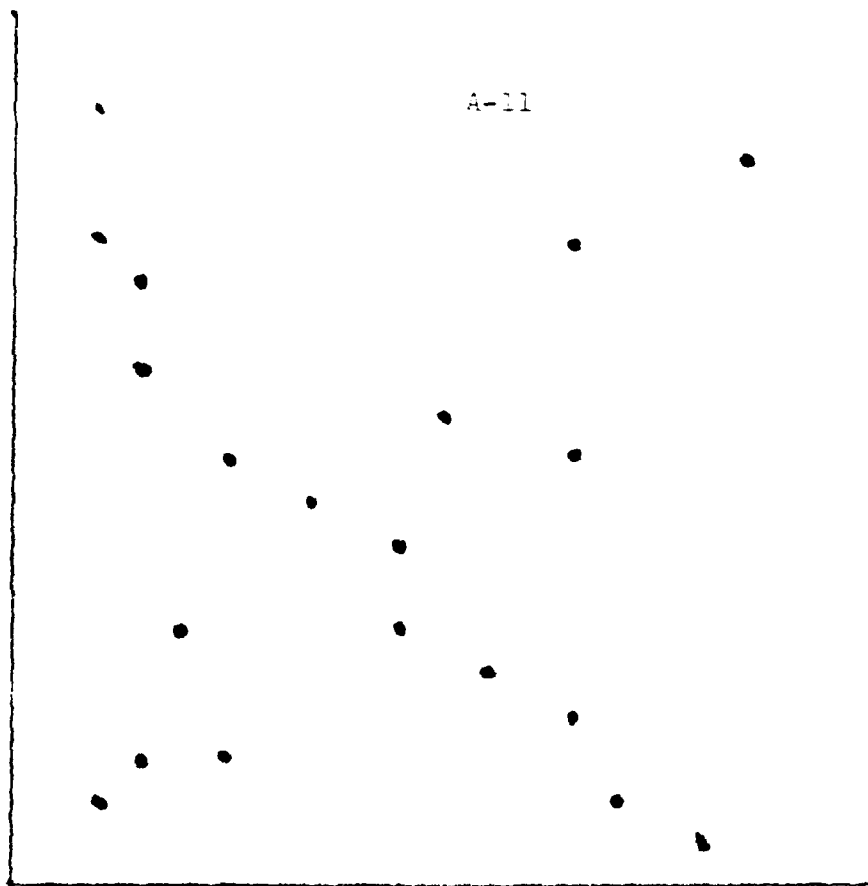
A-9



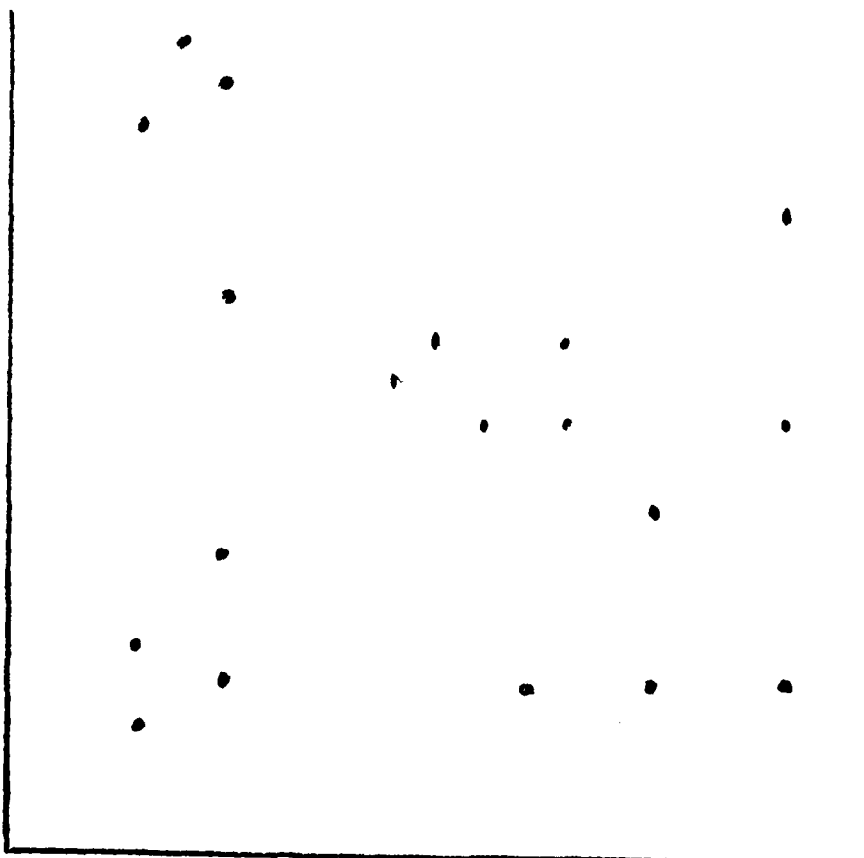
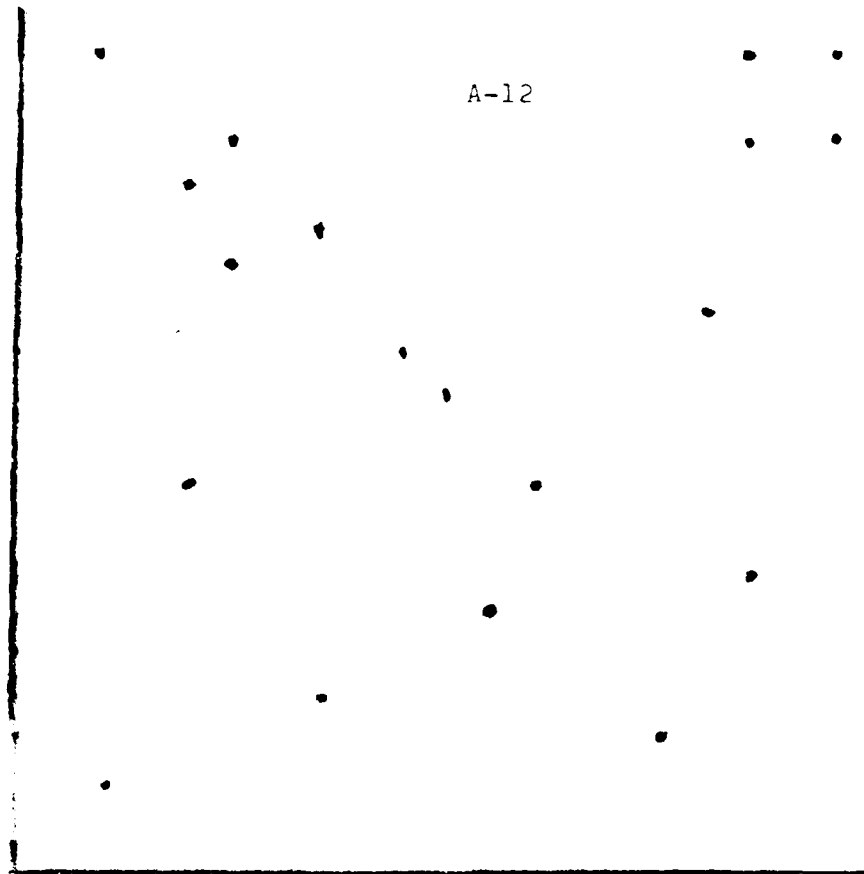
A-10



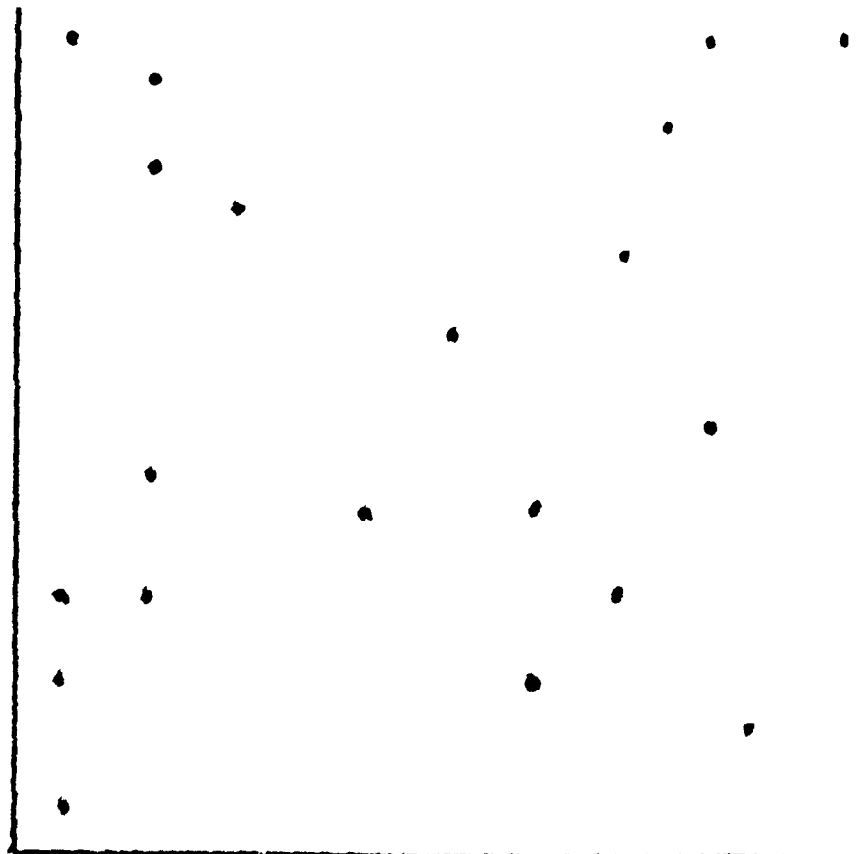
A-11



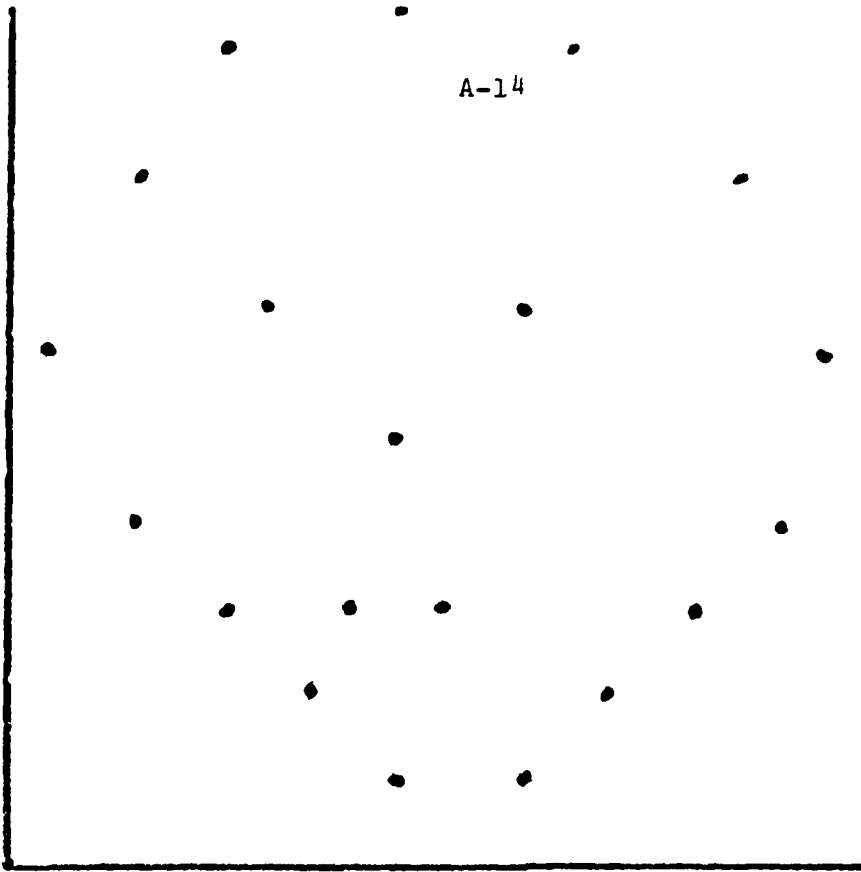
A-12



A-13

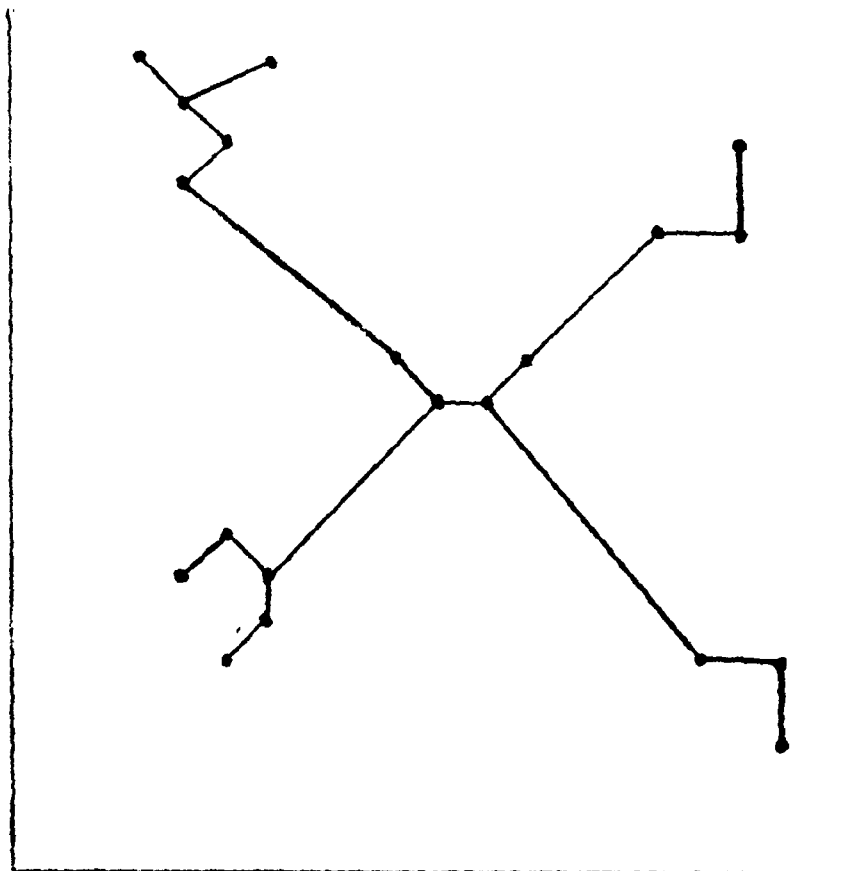
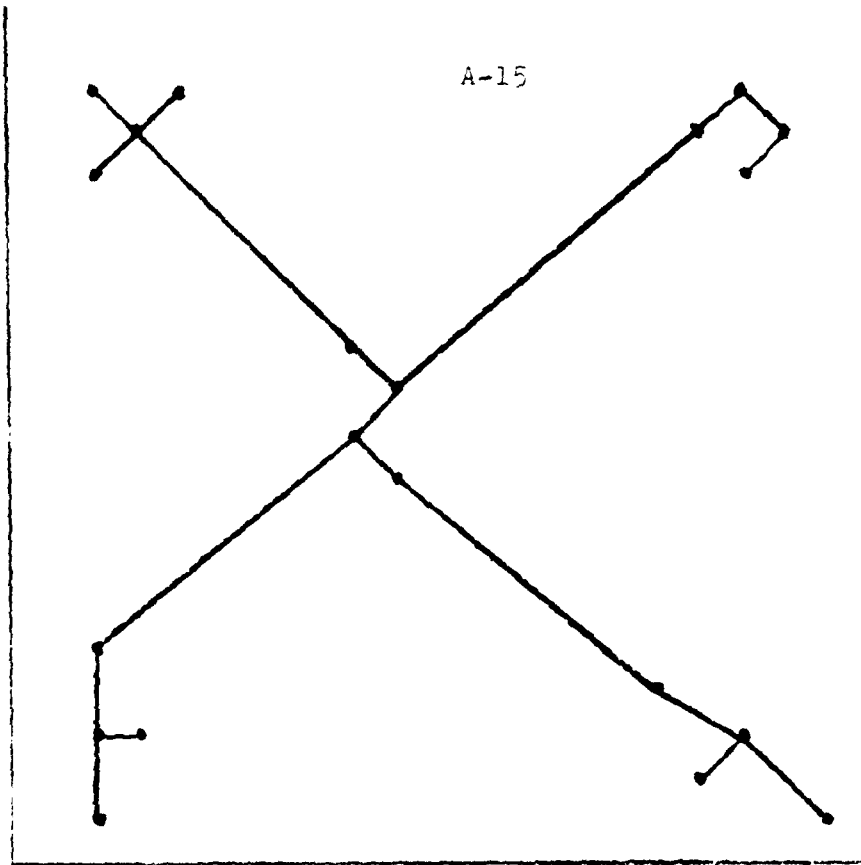


A-14

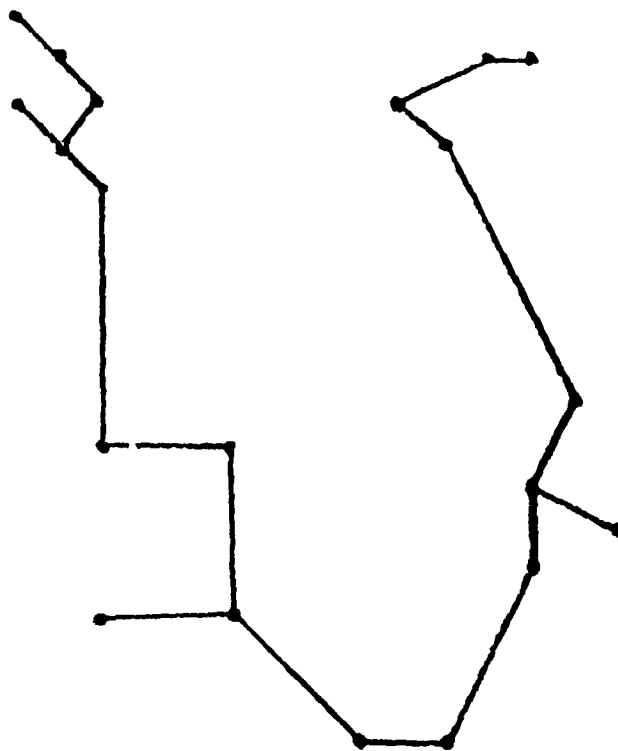
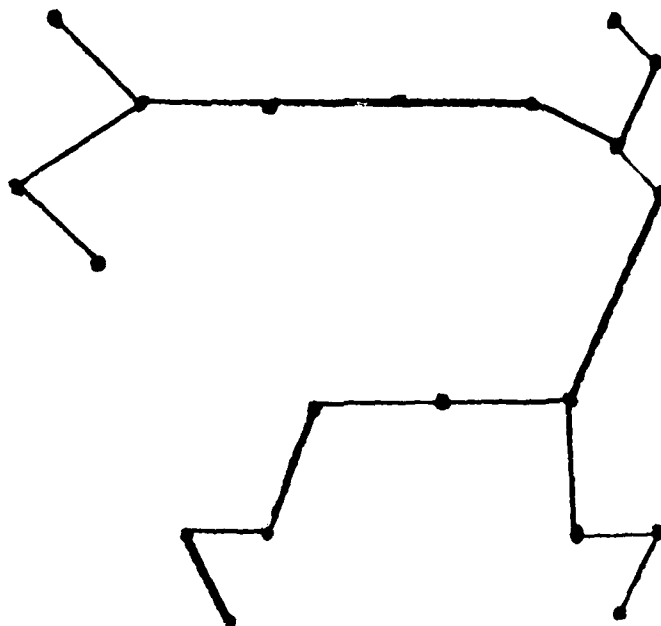


A-15

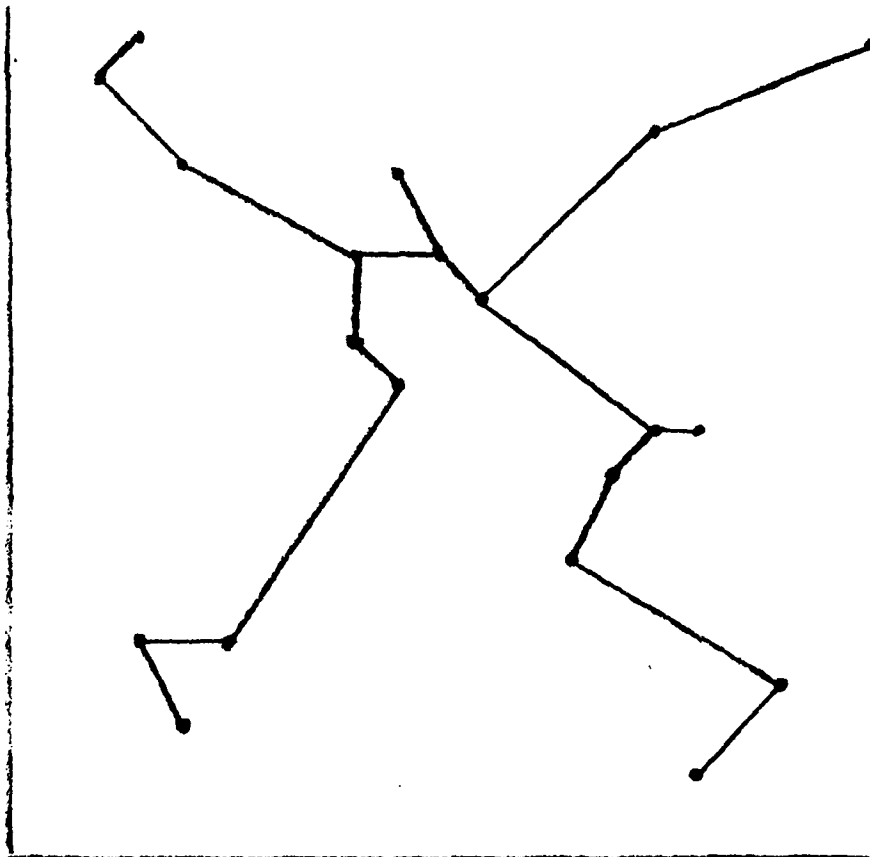
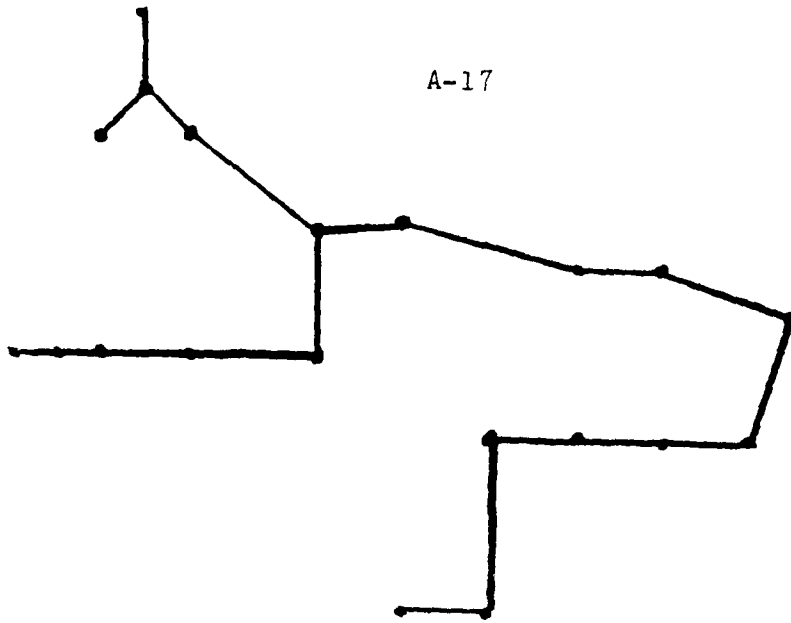
Appendix 3

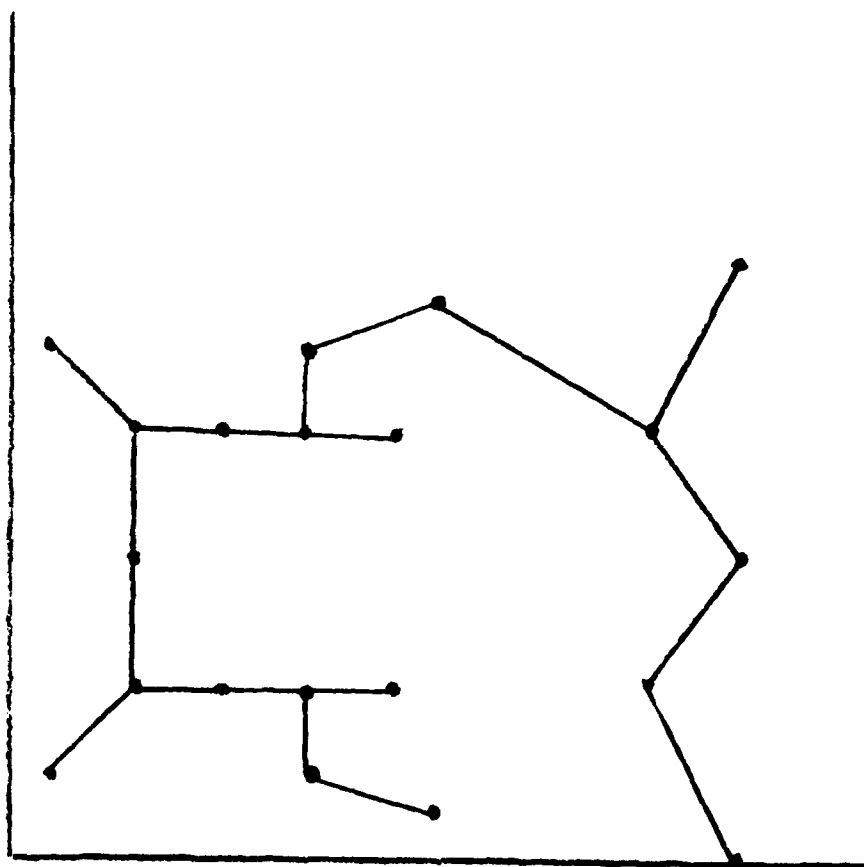
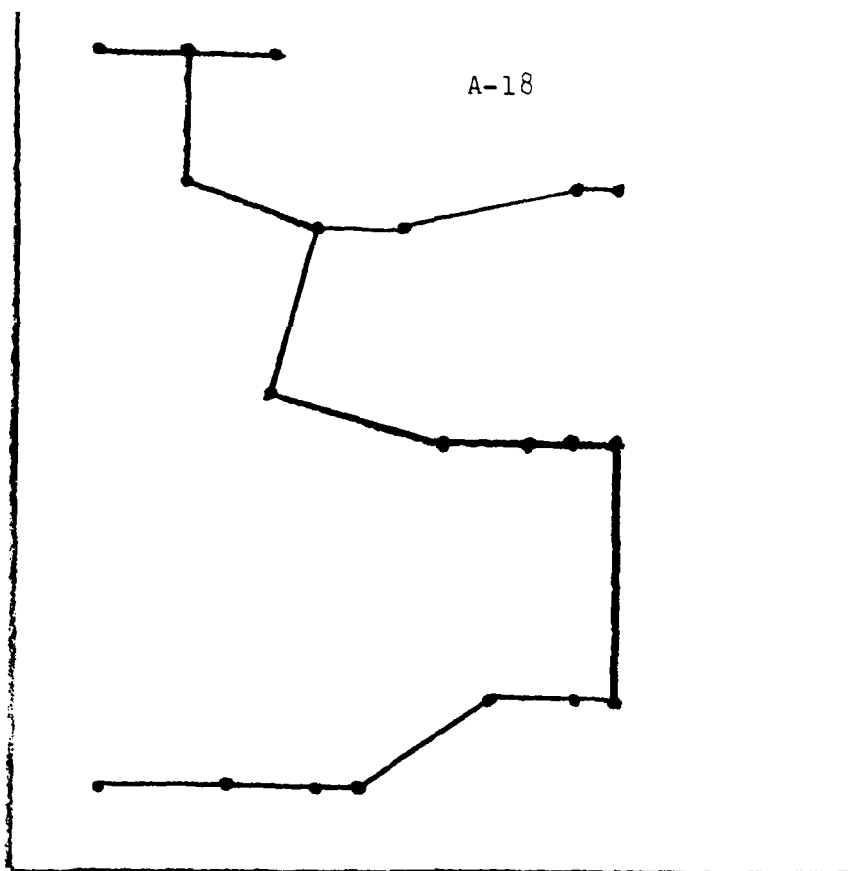


A-16

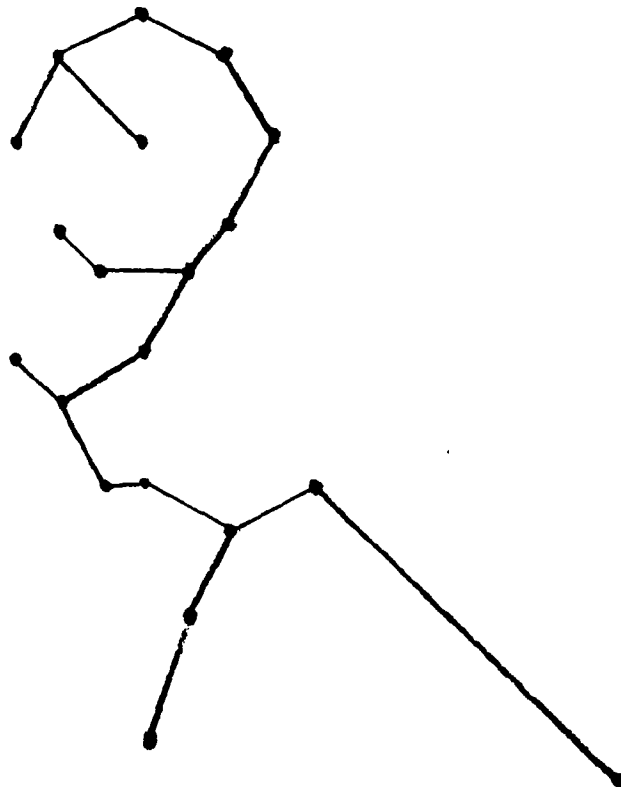
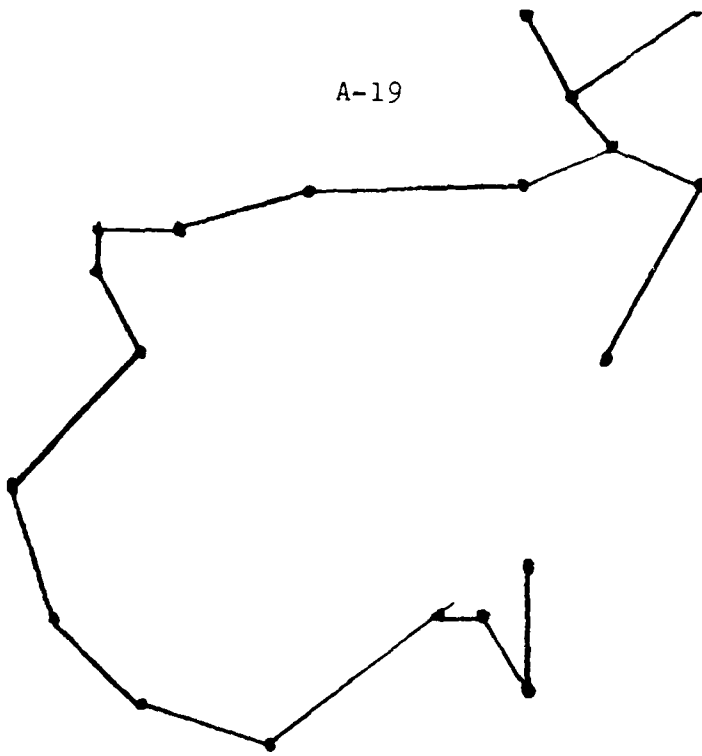


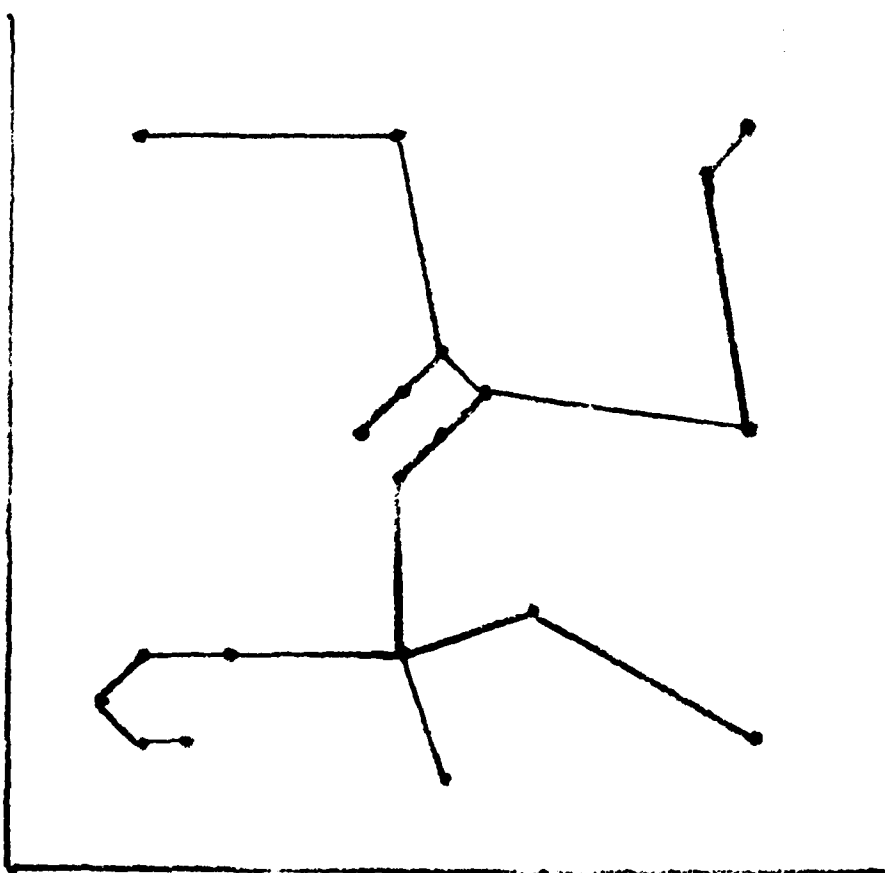
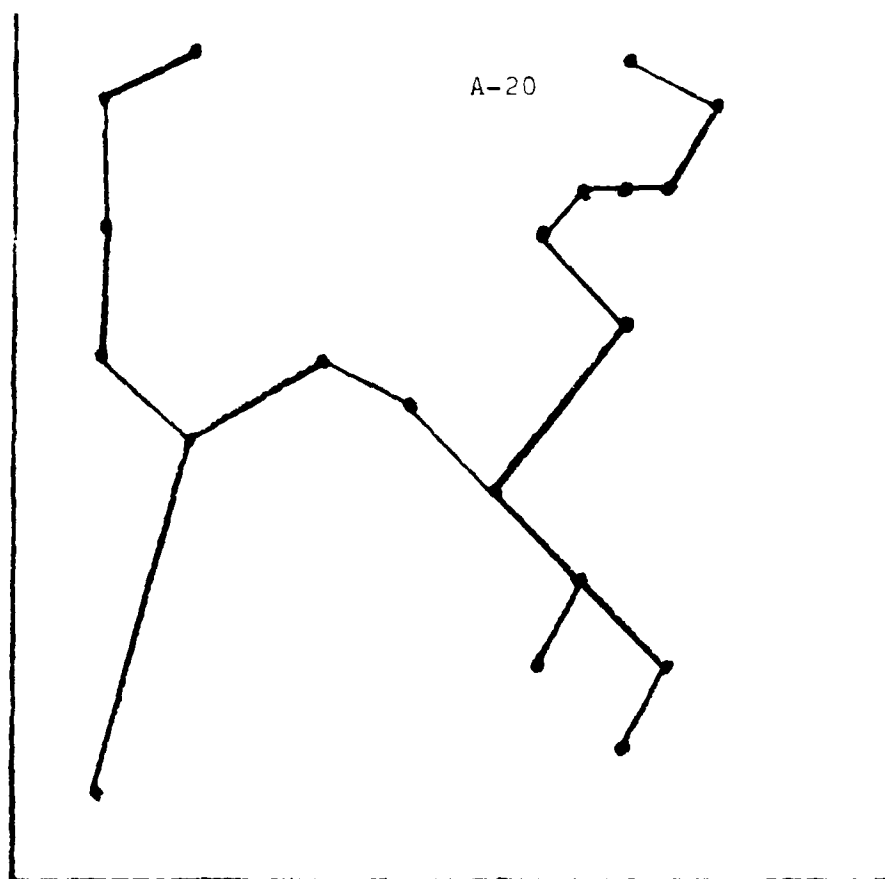
A-17



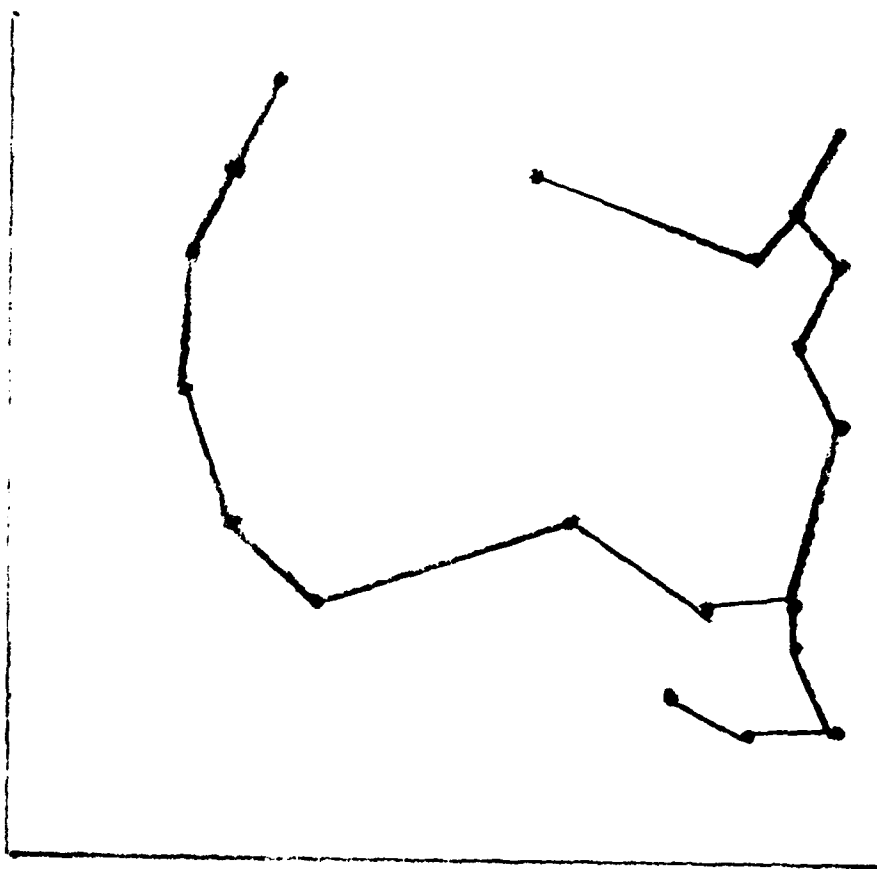


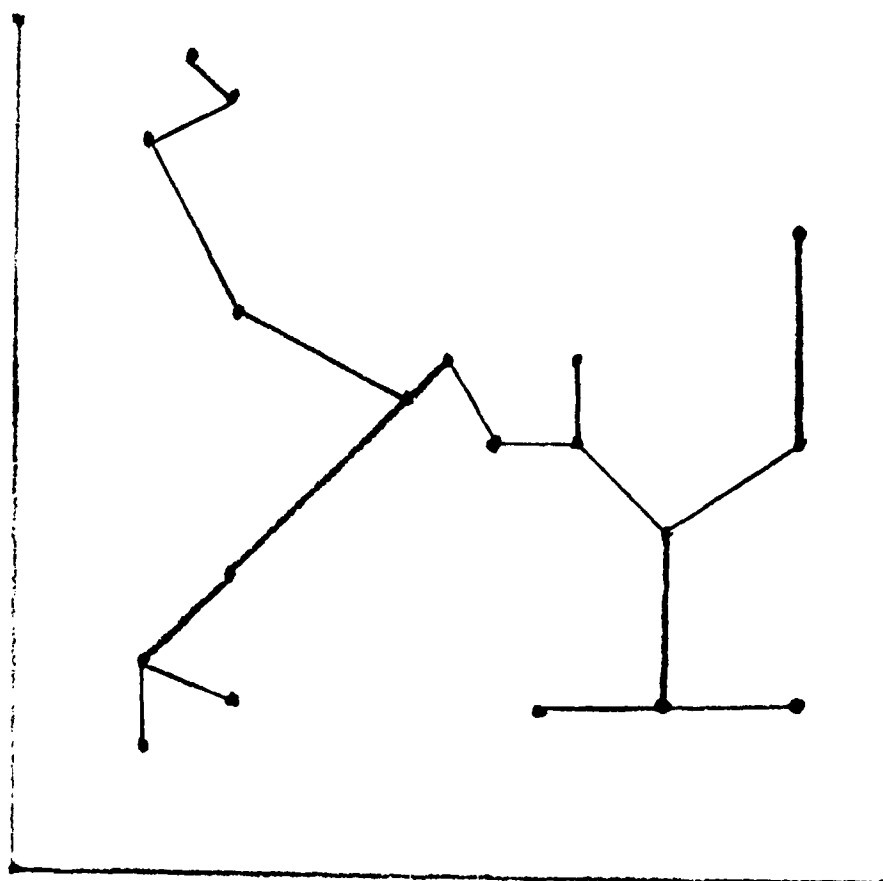
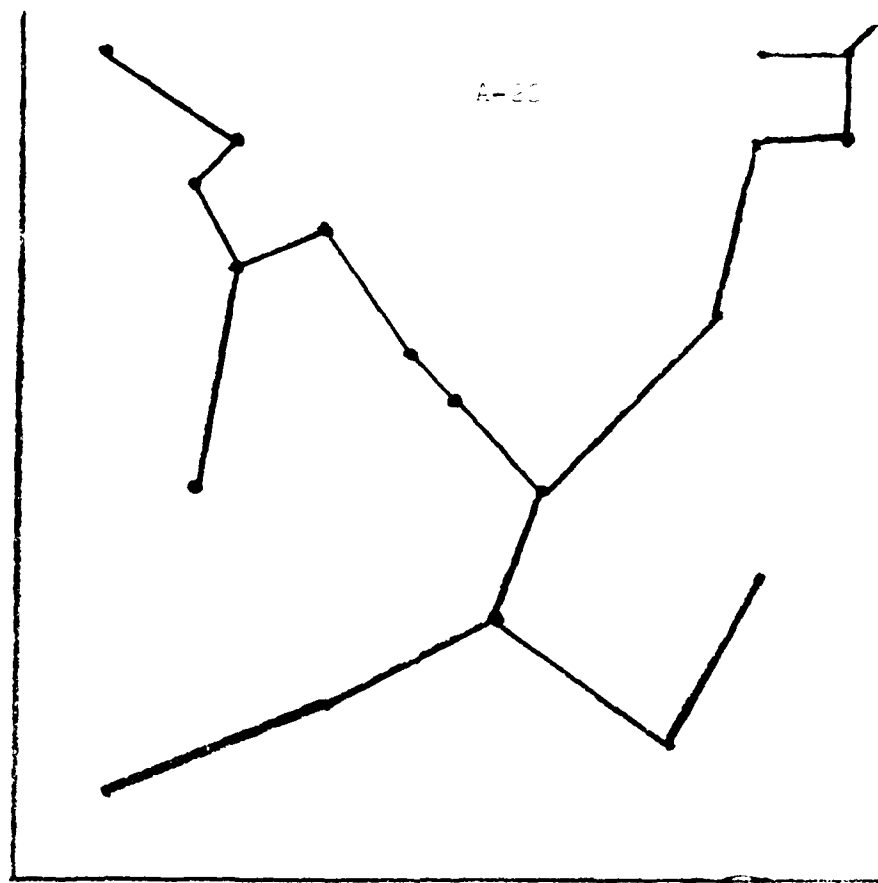
A-19

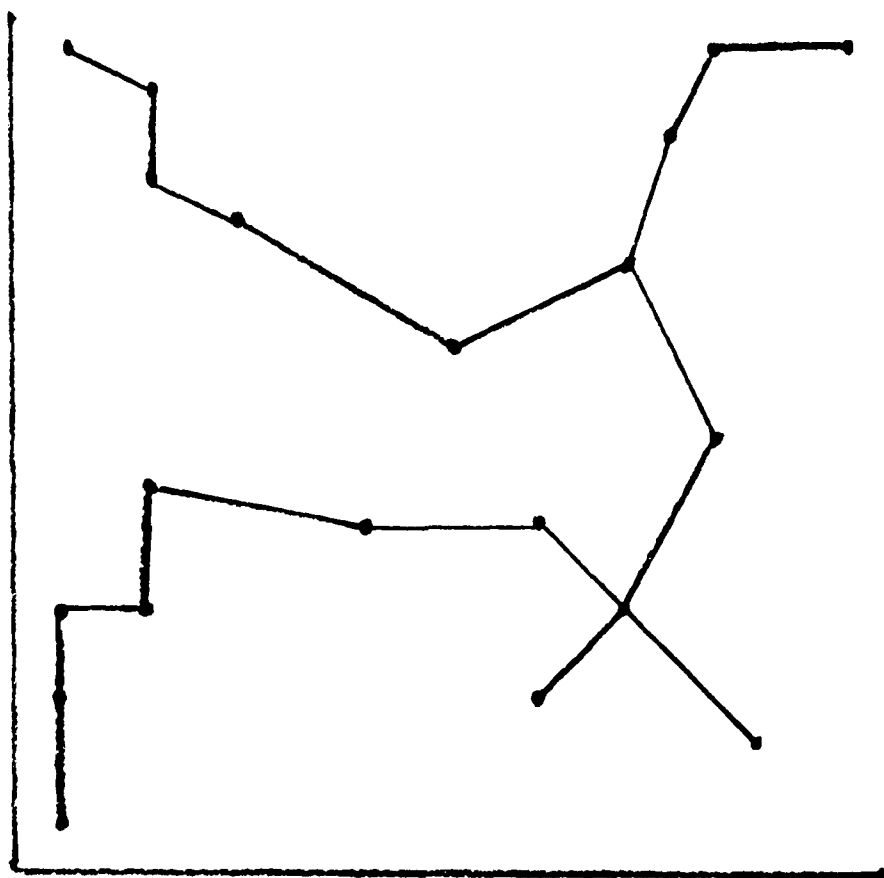
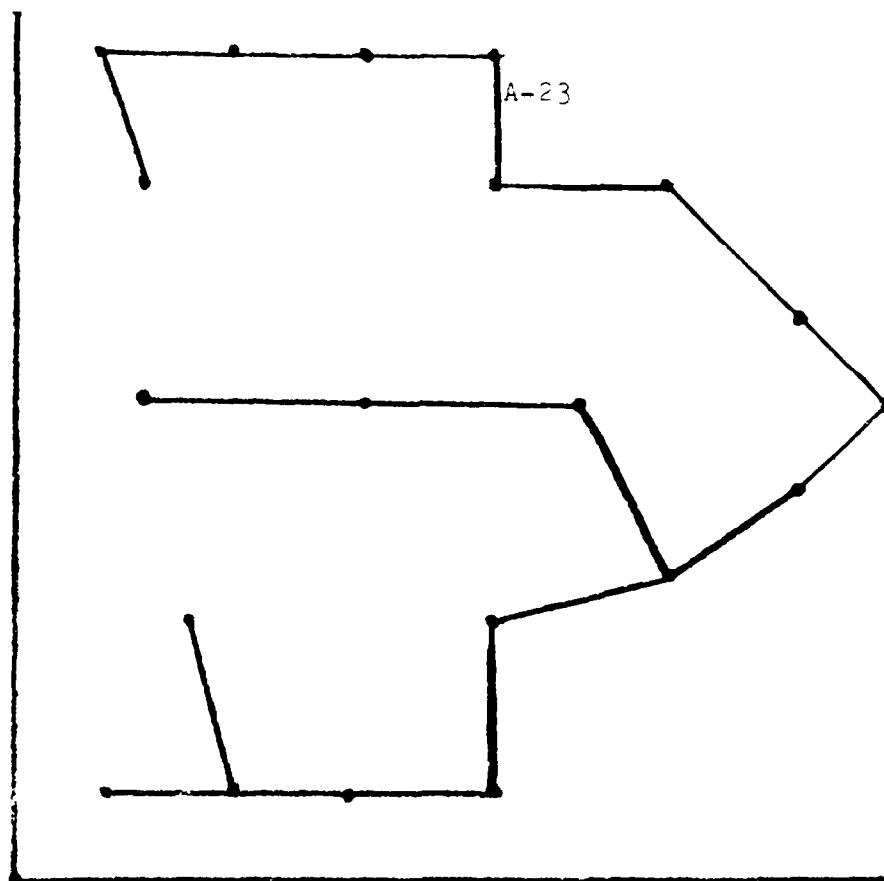


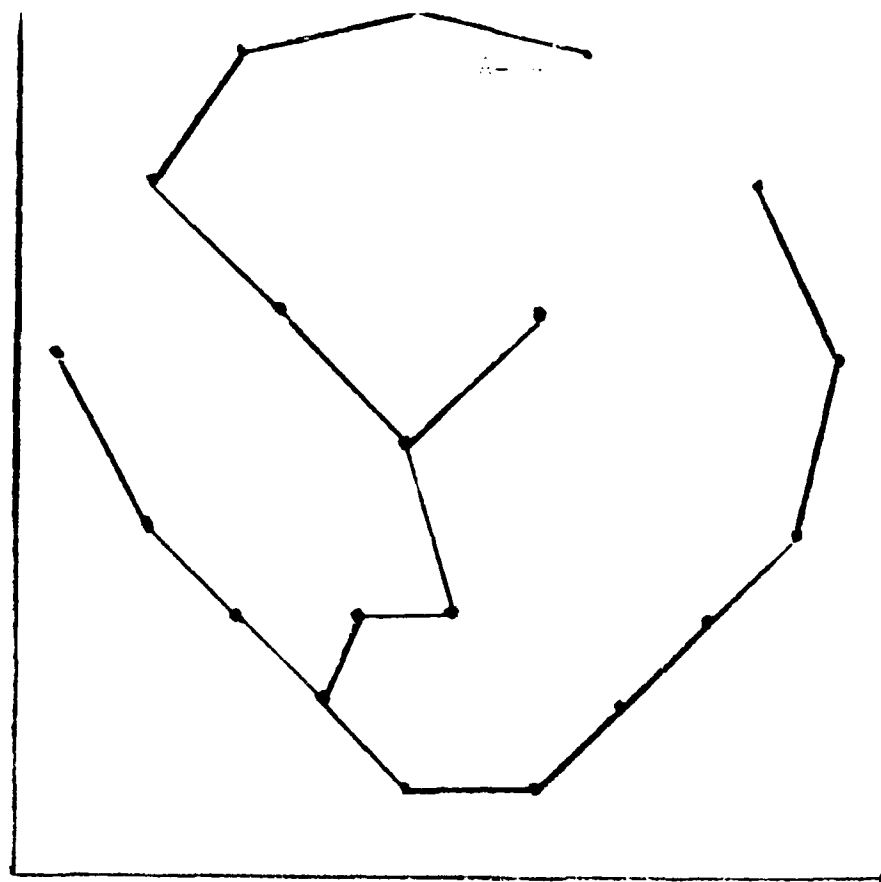


A-21









AD-A086 766

L N K CORP SILVER SPRING MD

F/G 5/8

PATTERN RECOGNITION INVESTIGATION AND ANALYSIS OF PLANAR POINT --ETC(U)

1979 D LAVINE, B LAMBIRD, L N KANAL

DAAK70-79-C-0091

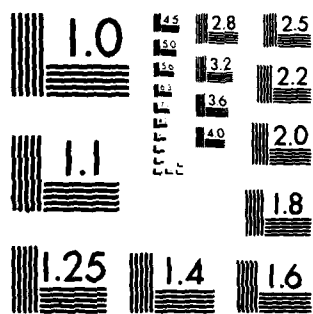
NL

UNCLASSIFIED

2 10 2

AL 0000 0000

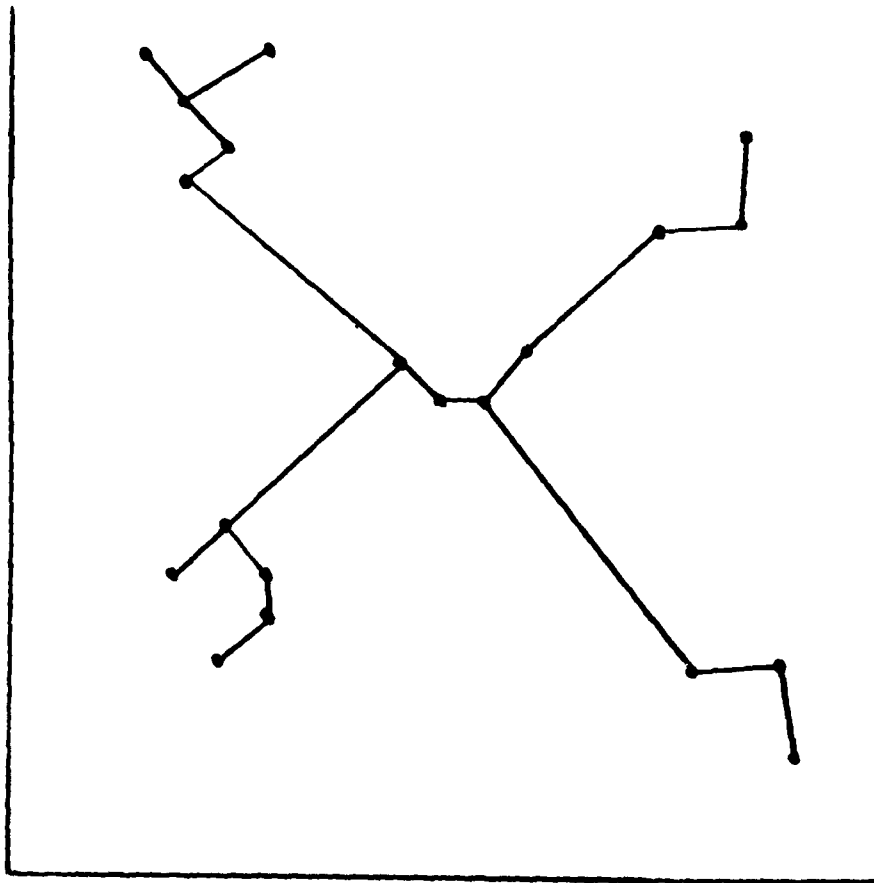
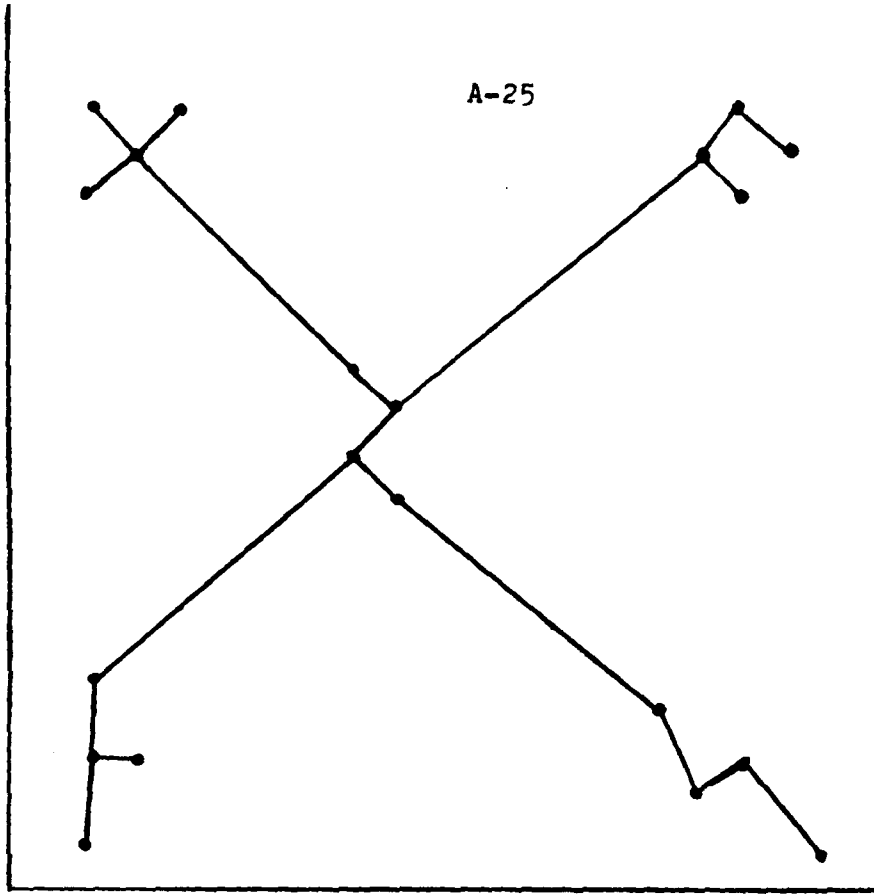
END
DATE
FILMED
8-80
DTIC



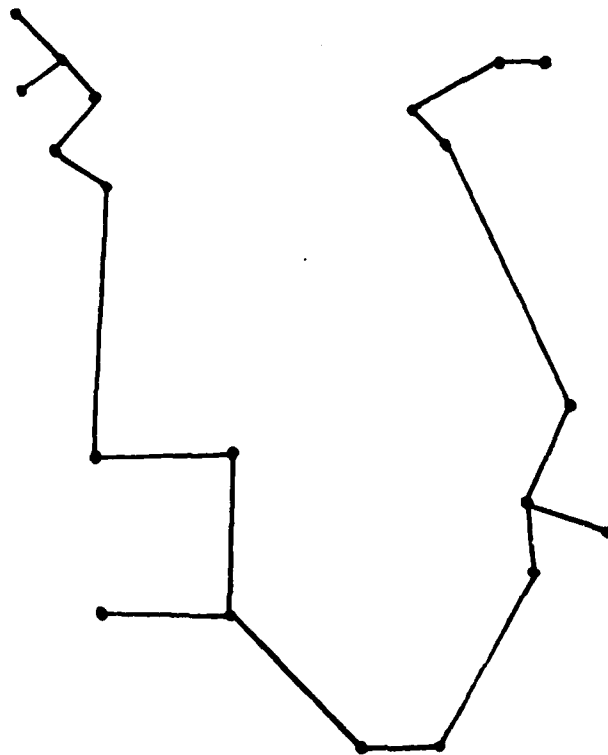
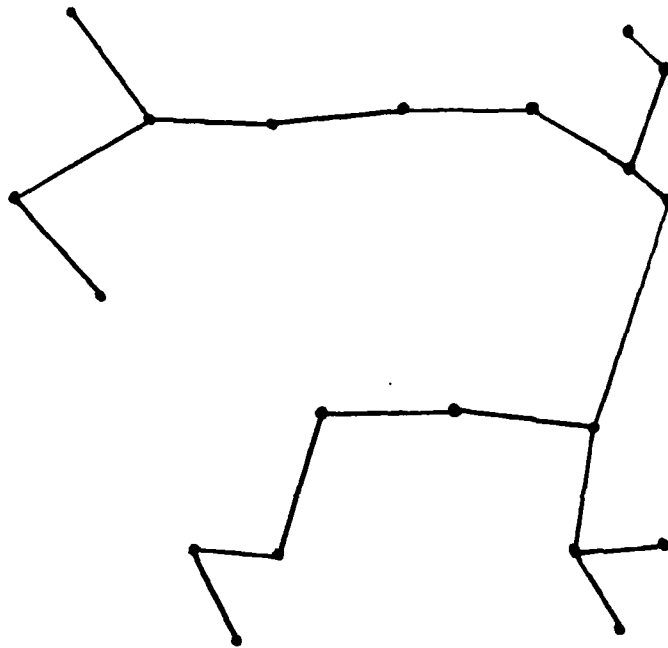
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

A-25

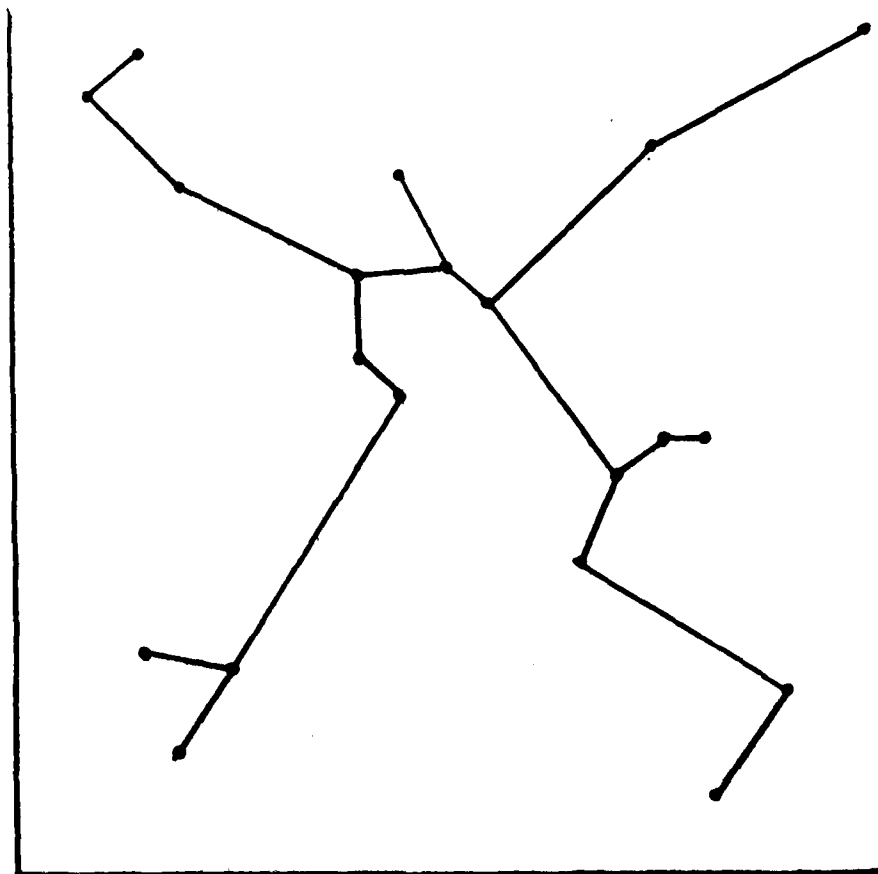
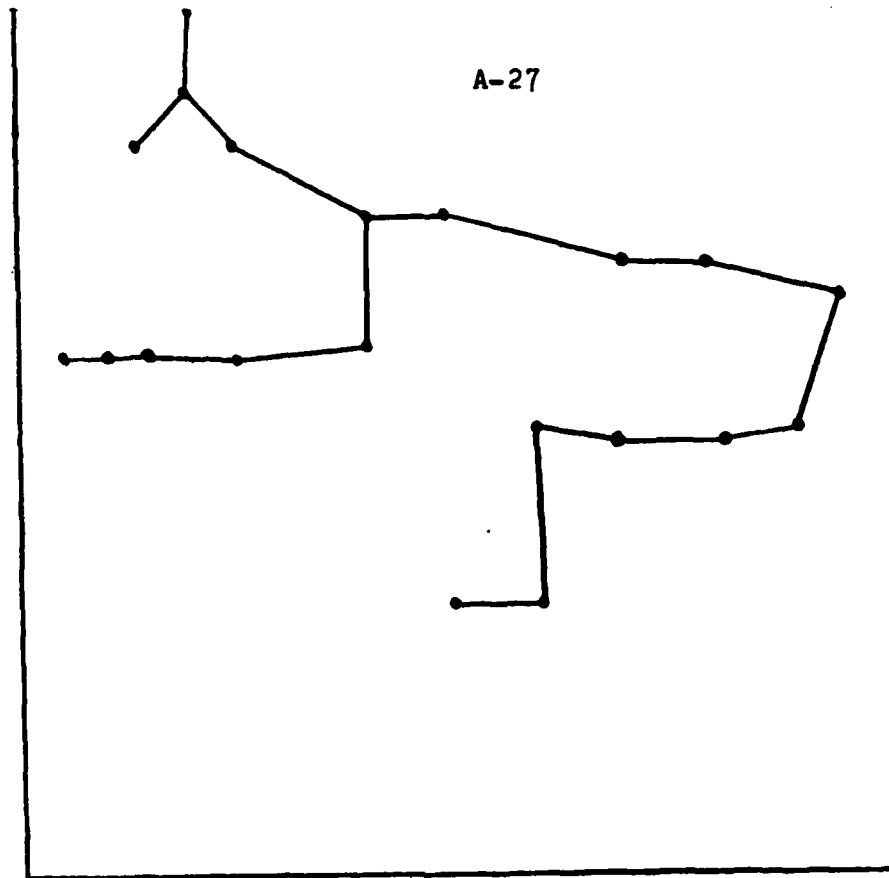
Appendix 4



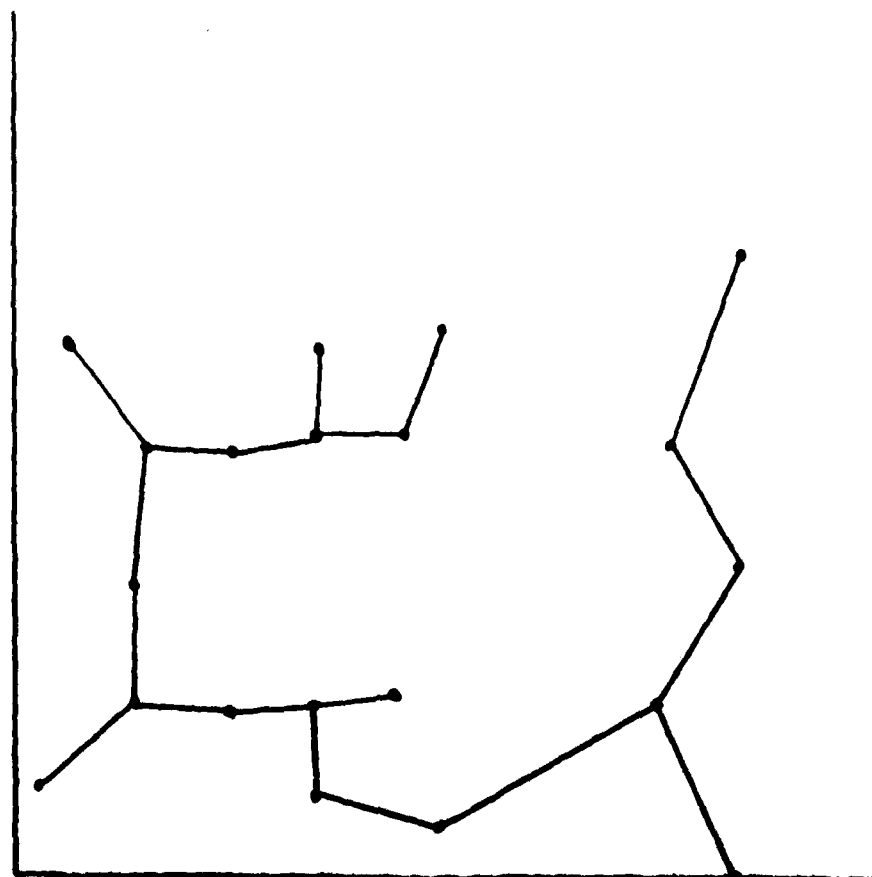
A-26

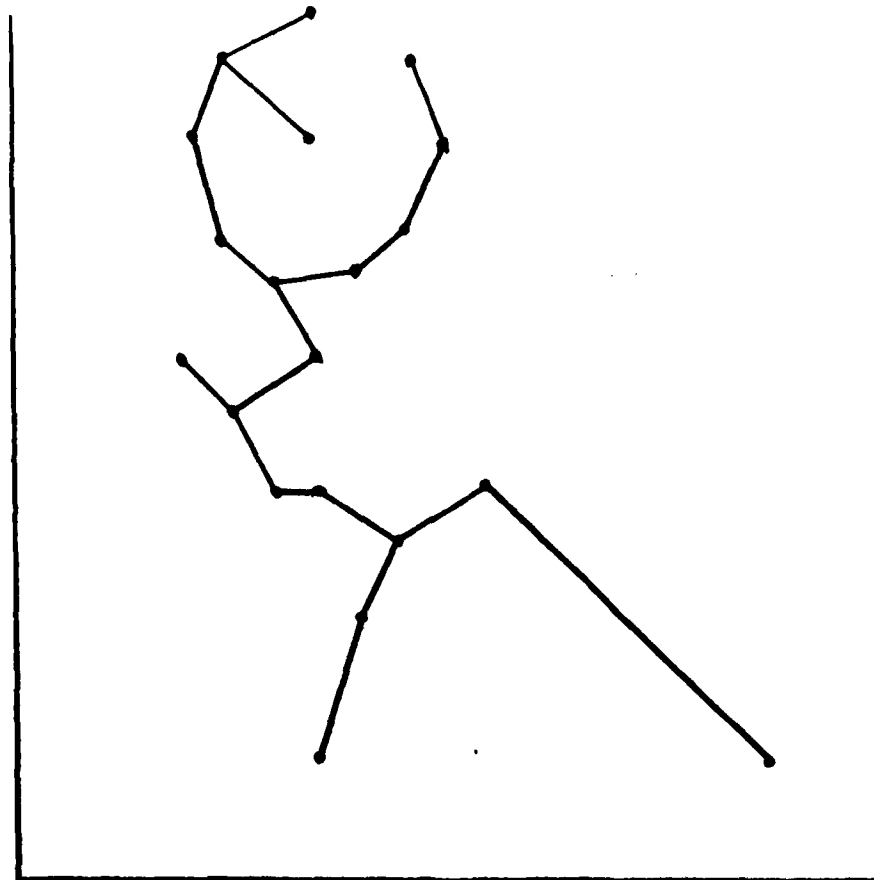


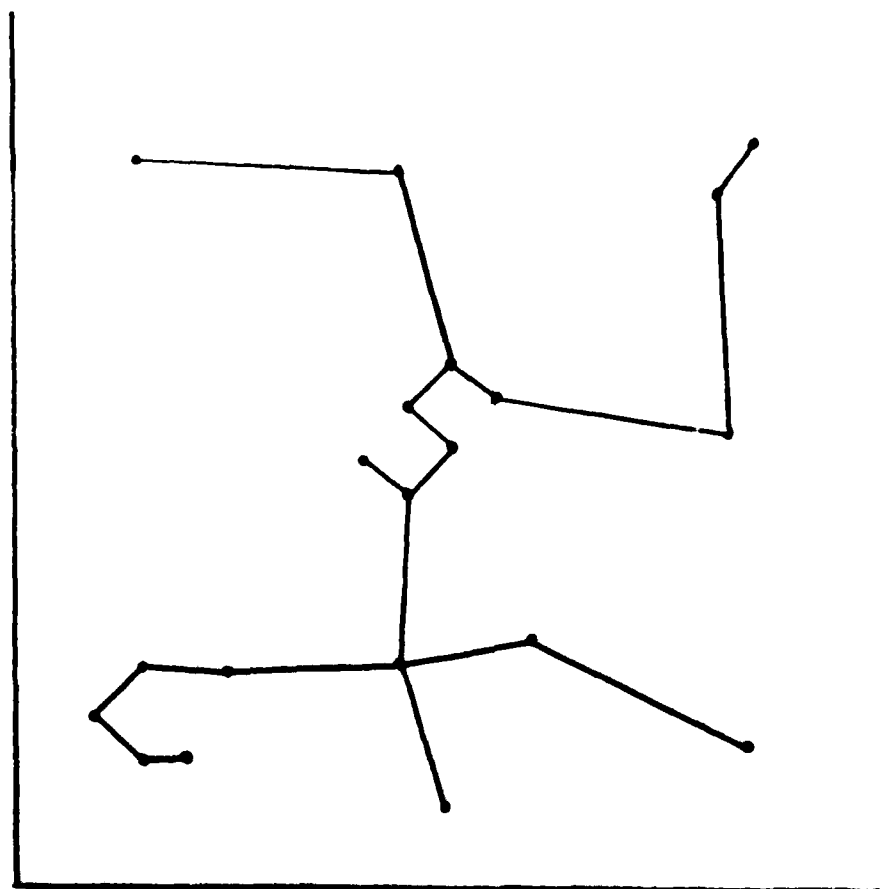
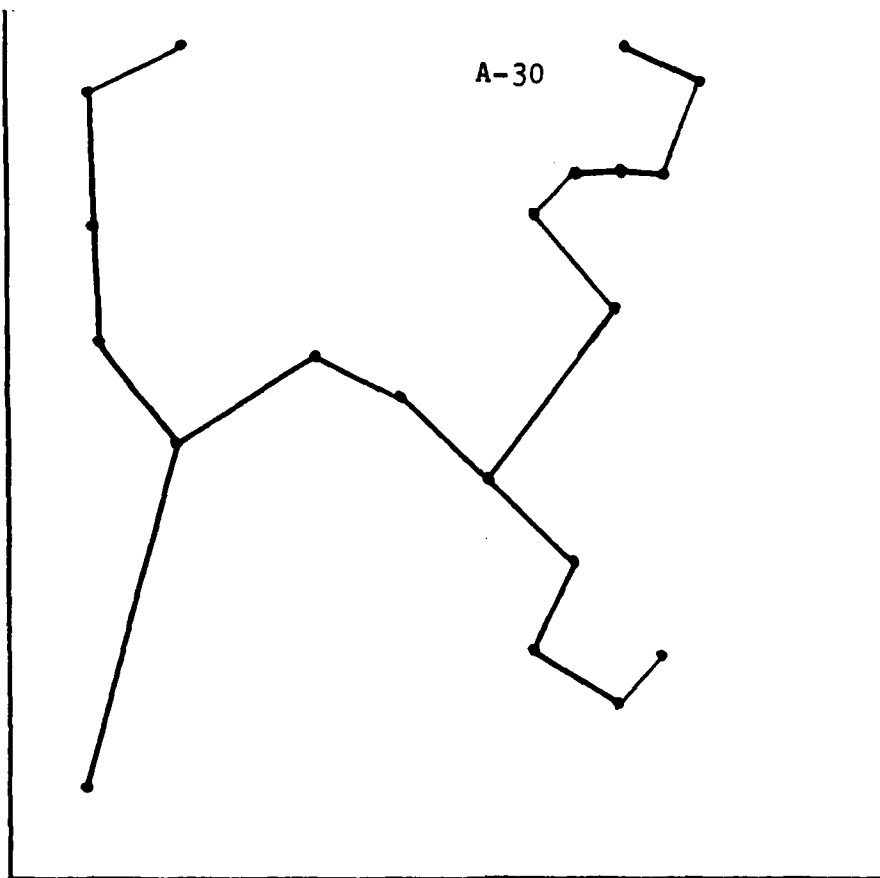
A-27



A-28

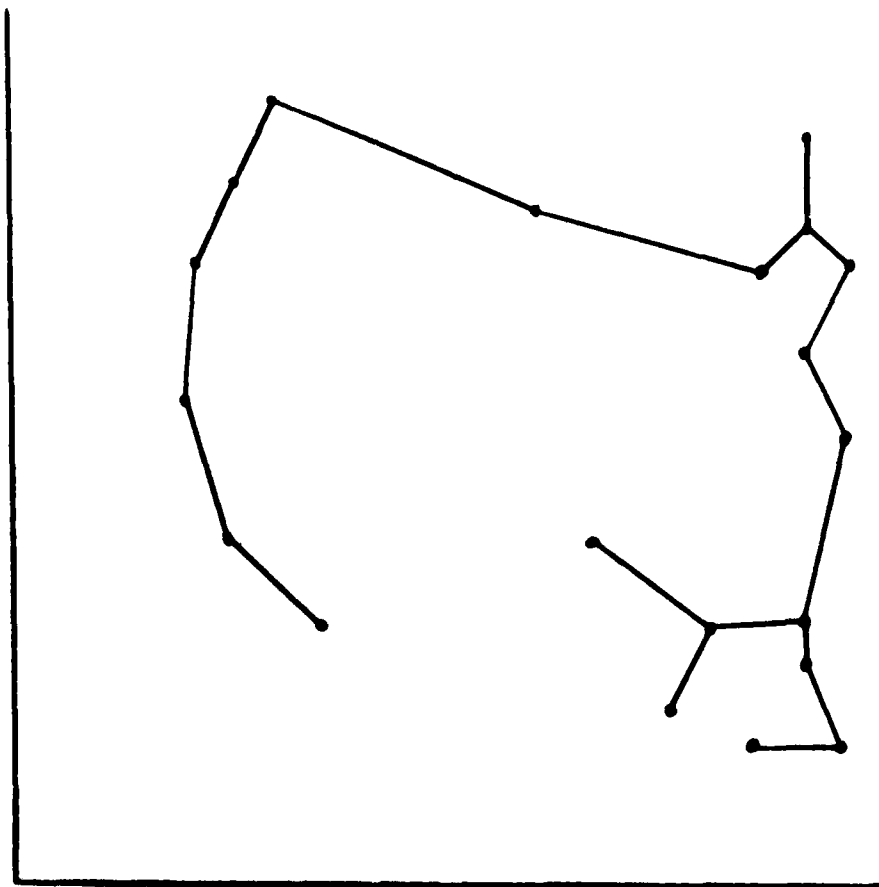


[illegible]

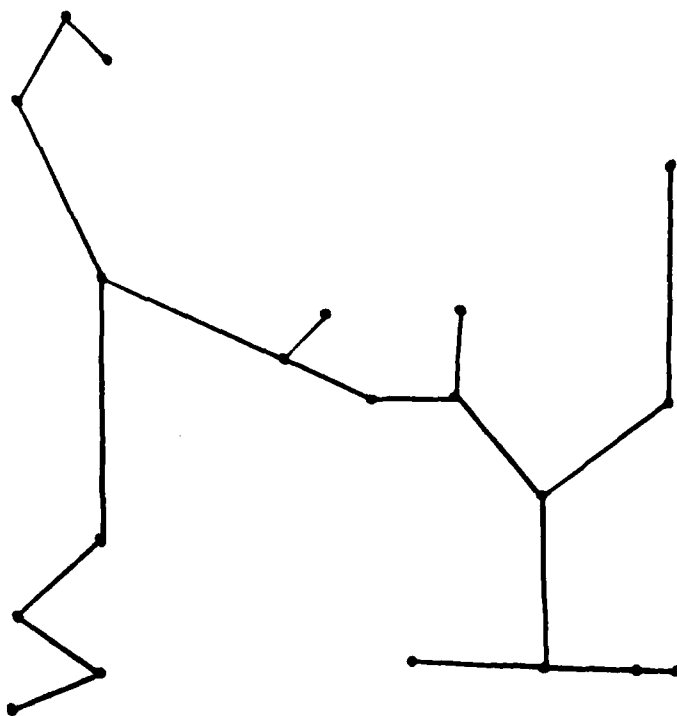
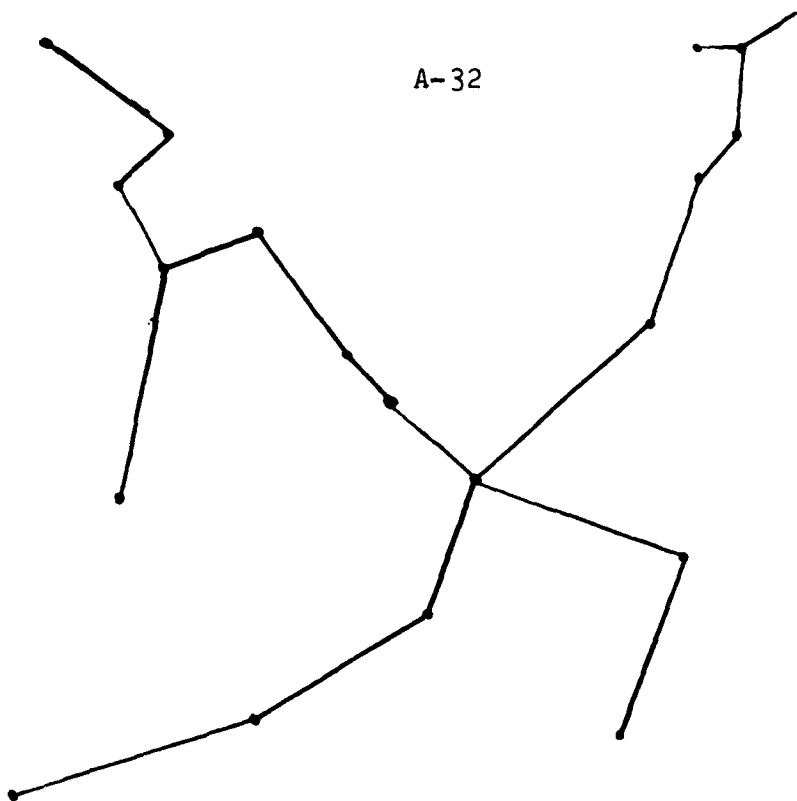


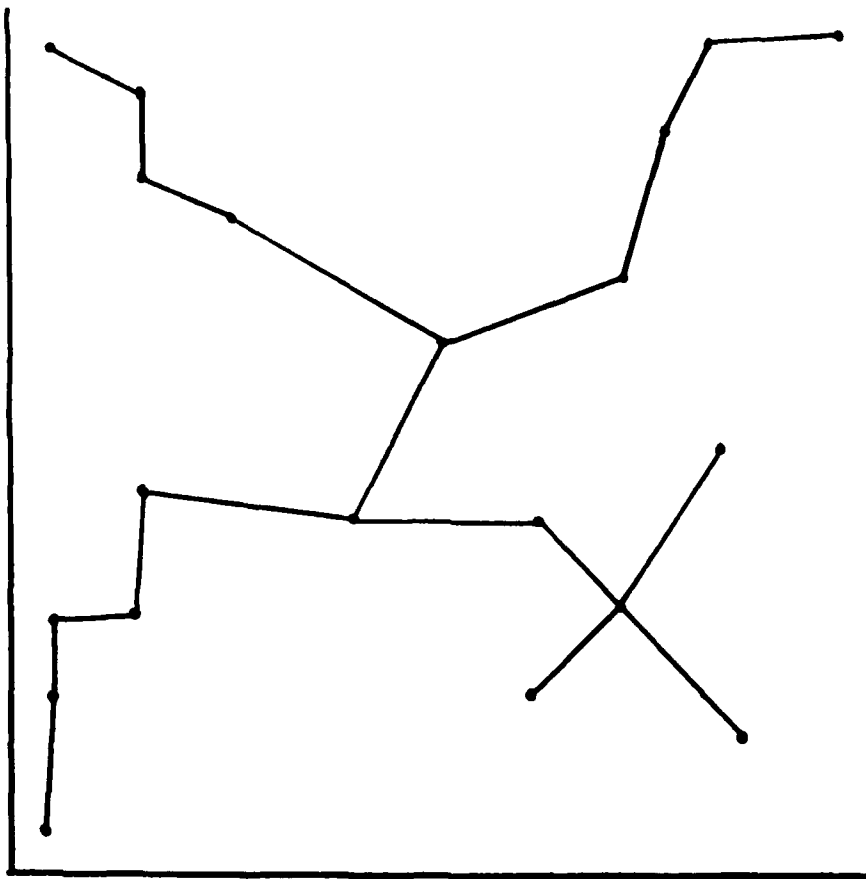
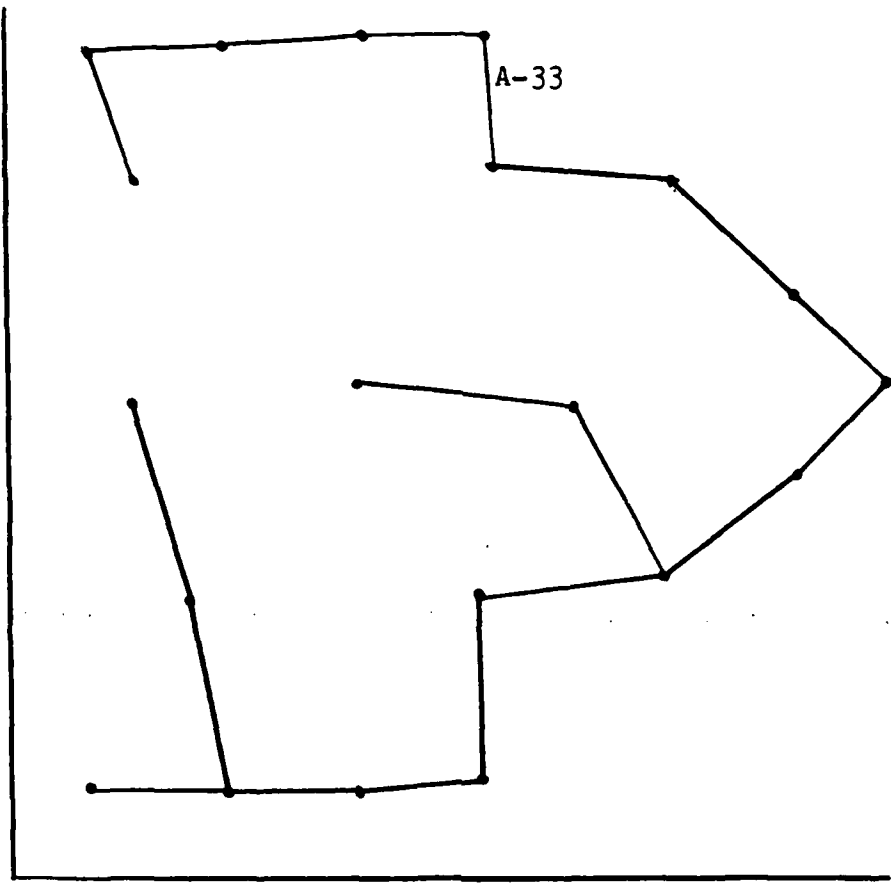
A-31

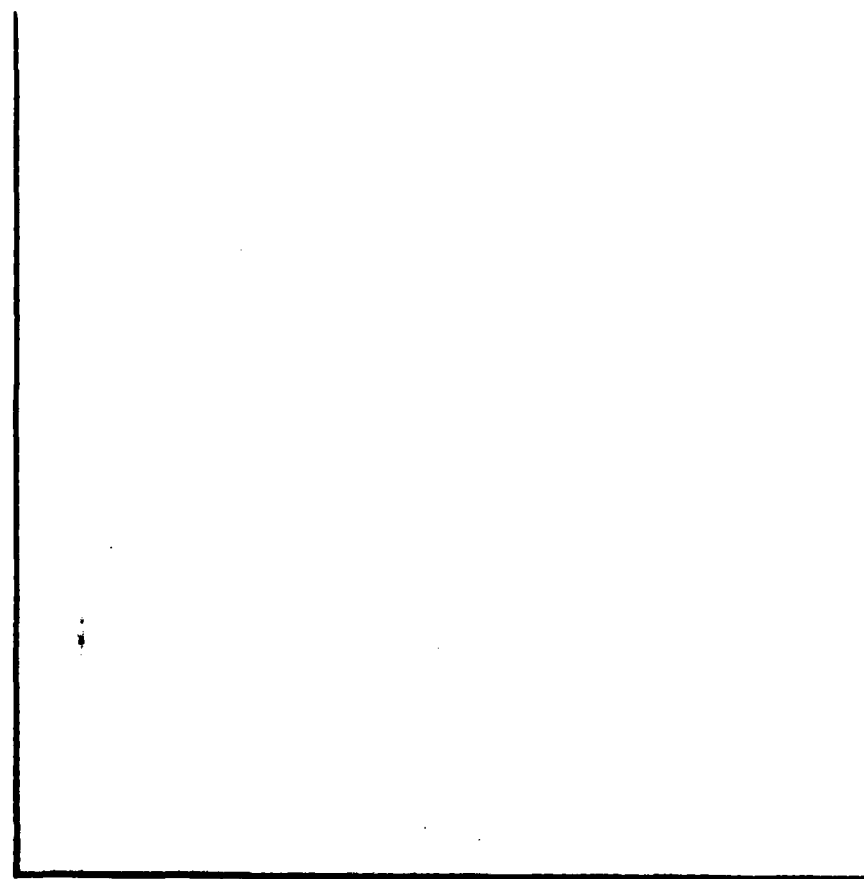
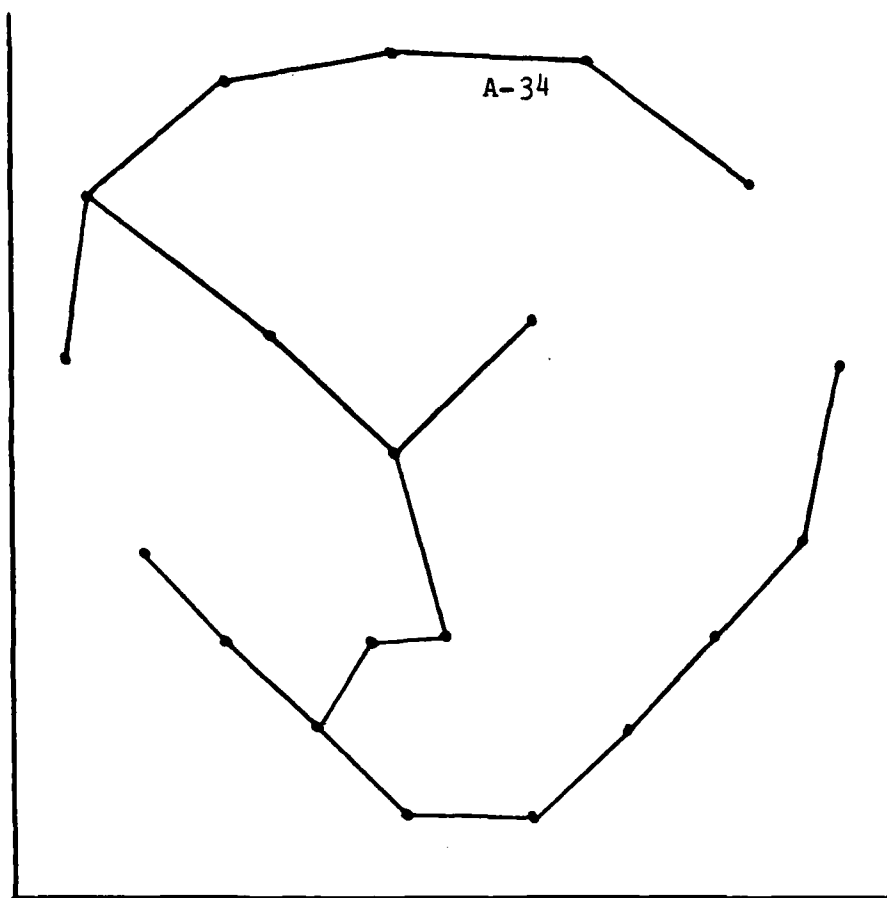
The graph shows a complex, branching path. It starts with a vertical segment on the left, then branches into several other segments. The path is composed of approximately 15 segments and 20 points. The overall shape is irregular and non-linear, with multiple turns and branches. The path starts at the top left, moves down, then branches into several other segments, some of which are nearly horizontal or vertical, while others are at various angles. The path ends at the bottom right.



A-32







DISTRIBUTION LIST

ARMY

Commander
US Army TRADOC
ATTN: ATCD-IE (COL W. H. Gardner)
Fort Monroe, VA 23651

Commander
US Army TCATA
ATTN: COL Stratton
Fort Hood, TX 76544

Commander
US Army TRADOC
ATTN: ATCD-M (LTC Webster)
Fort Monroe, VA 23651

Commander
US Army MICOM
ATTN: DRSMI-RAS (Mr. J. Hagood)
Redstone Arsenal
Huntsville, AL 35809

Commander
US Army TRADOC
ATTN: Dave Ingram
Fort Monroe, VA 23651

Commander
US Army MICOM
ATTN: DRDMI-TEL (H. Tracy Jackson)
Redstone Arsenal, AL 35809

Commander
US Army TRADOC
ATTN: ATDOC-TA (Dr. Marvin Pastel)
Fort Monroe, VA 23651

Commander
US Army Foreign Science and
Technology Center
ATTN: DRXST-SD (George Riley)
220 Seventh Street
Charlottesville, VA 22901

Commander US Army DARCOM
ATTN: MG S. R. Sheridan
Director of Development
and Engineering
5001 Eisenhower Avenue
Alexandria, VA 22333

Commandant
US Army Engineer School
ATTN: Director of Combat Developments
Fort Belvoir, VA 22060

Commander
US Army DARCOM
ATTN: DRXSP (B. Chasnov)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commandant
US Army Air Defense School
ATTN: MG John B. Oblinger, Jr.
Fort Bliss, TX 79916

Commander
US Army HQ DARCOM
ATTN: DRCDE-DS (Mr. Robert McGowan)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commandant
US Army Air Defense School
ATTN: ATSA-TSM-P (COL Sam Hayton)
Fort Bliss, TX 79916

ARMY (Continued)

Commander
US Army ERADCOM
Harry Diamond Laboratories
ATTN: DRDEL-AP-CCM (COL Murray)
2800 Powder Mill Road
Adelphia, MD 20793

Director
US Army Night Vision and Electro-
Optical Laboratories
ATTN: John Johnson
Fort Belvoir, VA 22060

Commander
US Army TARADCOM
ATTN: DRDTA-ZS (Otto Renius)
Warren, MI 48090

Director
US Army Night Vision and Electro-
Optical Laboratories
ATTN: Frank Shield
Fort Belvoir, VA 22060

Commander
US Army White Sands Missile Range
ATTN: Dr. Duncan, Tech Dir.
White Sands Missile Range, NM 88002

Director
US Army Night Vision and Electro-
Optical Laboratories
ATTN: Russ Moulton
Fort Belvoir, VA 22060

Commander
USAINSCOM Pentagon Counterintelligence
Force (Room BE800)
ATTN: Mr. Thomas L. Forman
The Pentagon
Washington, D.C. 20310

Commander
US Army Mobility Equipment Research
and Development Command
ATTN: DRDME-RT (Dr. John W. Bond)
Fort Belvoir, VA 22060

HQ DA
ATTN: DAMI-CI (COL D. Fink)
Washington, D. C. 20310

Commander
US Army Mobility Equipment Research
and Development Command
ATTN: DRDME-WC (Vault)
Fort Belvoir, VA 22060 (2 copies)

Project Manager
PATRIOT Missile System, DARCOM
ATTN: DRCPM-MD-T-T (P. Bingham)
Redstone Arsenal, AL 35809

Commander
US Army Combined Arms Center
and Fort Levenworth
ATTN: ATZLCA-COG (David Bitters)
Fort Levenworth, KS 66027

AIR FORCE

Commander
USAF Systems Command
ATTN: AFSC/SD (MG McMullen)
Andrews AFB, MD 20334

Commander
USAF Systems Command
ATTN: AFSC/XR (BG Jacobs)
Andrews AFB, MD 20334

Commander
USAF Systems Command
ATTN: AFSC/SDW (COL Robert Bannach)
Andrews AFB, MD 20334

Commander
USAF Systems Command
ATTN: AFSC/INA (CPT Tommy L. Brazie)
Andrews AFB, MD 20334

Commander
USAF Tactical Air Command
ATTN: LGM (COL Sauls)
Langley AFB, VA 23665

Commander
USAF Weapons Development Center
ATTN: Dr. Joe Mayersak
Eglin AFB, FL 32542

Commander
Foreign Technology Division, AFSC
ATTN: FTD/SDEC (Mr. Robert C. Frick)
Wright-Patterson AFB, OH 45433

Commander
AFSC/YAD
ATTN: CAPT W. Moomey
P. O. Box 92962
Worldway Postal Center
Los Angeles, CA 90009

Commander
AFSC
ATTN: ASD/AERA (LTC Lambiase)
Wright Patterson AFB, OH 45433

Commander
AFSC/YAV
ATTN: COL T. May
P. O. Box 92962
Worldway Postal Center
Los Angeles, CA 90009

Commander
Air Force Geophysics Laboratory/OPR
ATTN: Mr. B. Sandford
L. G. Hanscom Field
Bedford, MA 01731

Commander
Rome Air Development Center
ATTN: RADC/IRRE (Mr. Paul Smith)
Griffis Air Force Base, NY 13441

Commander
Rome Air Development Center
ATTN: RBCA (Jack Edwards)
Griffis Air Force Base, NY 13441

Commander
Rome Air Development Center
ATTN: RADC/OCDR (Doug Lynch)
Griffis Air Force Base, NY 13441

Commander
Air Force Avionics Laboratory
ATTN: Paul Huffman
Wright-Patterson AFB, OH 45433

Commander
Air Force Avionics Laboratory
ATTN: ASD/AER (Mr. Merle Carr)
Wright Patterson AFB, OH 45433

Commander
Air Force Avionics Laboratory
ATTN: AFAL/RWI (Harold Watson)
Wright Patterson AFB, OH 45433

NAVY

Commander
Naval Research Laboratory
ATTN: Dr. J. MacCullum
4555 Overlook Avenue, SW
Washington, D.C. 20375

Commander
US Naval Air Systems Command
ATTN: CPT Donald N. Jarris
(Code NAIR 547)
Washington, D.C. 20361

Commander
US Naval Air Systems Command
ATTN: Pete Luppino (Code NAIR 360D)
Washington, D. C. 20361

Commander
Naval Weapons Center
ATTN: Stuart Briel (Code 3943)
China Lake, CA 93555

Commander
Naval Electronics Systems Command
ATTN: Mr. Albert D. Ritter
(DME-107-52A)
Washington, D.C. 20360

HQ US Marine Corps
Navy Annex
ATTN: Dr. Alex L. Slafkosky
(MC-RD-1, Code ABQ, Room 2107)
Washington, D. C. 20390

DEPARTMENT OF DEFENSE

OUSDR&E
ATTN: Mr. Robert Moore
The Pentagon, Room 3E1044
Washington, D. C. 20301

OUSDR&E
ATTN: Mr. D. Brockway
The Pentagon
Washington, D. C. 20301

Director
Defense Advanced Research
Projects Agency
ATTN: Dr. R. Fossum
1400 Wilson Blvd.
Arlington, VA 22209

Director
Defense Advanced Research
Projects Agency
ATTN: Dr. J. Tegnalia
1400 Wilson Blvd.
Arlington, VA 22209

Director
Defense Advanced Research
Projects Agency
ATTN: Dr. S. Zakanycz
1400 Wilson Blvd.
Arlington, VA 22209

USDRE/OD (Land Warfare)
ATTN: Mr. K. R. Sturdivant
Room 3E1025, The Pentagon
Washington, D. C. 20301

DEPARTMENT OF DEFENSE (Continued)

Director
Central Intelligence Agency
ATTN: R. E. Lawrence, ORD
Washington, D. C. 20505

Commander
Defense Intelligence Agency
ATTN: DIA/DC-7B
(LTC Samuel L. Stille)
The Pentagon
Washington, D.C. 20301

Director
National Security Agency
ATTN: Mr. R. T. Thompson
Fort George G. Meade, MD 20755

Joint Cruise Missile Project Office
ATTN: JCM-G-70 (LTC Humphreys)
Washington, D.C. 20361

Director
Defense Nuclear Agency
ATTN: COL Jackson
Washington, D. C. 20305

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314 (2 copies)

PRIVATE INDUSTRY - PROFIT

Manager, Business Department Analytic
ATTN: Donald S. Holmes
Systems & Operations Division
2500 Maryland Road
Willow Grove, PA 19090

General Research Corporation
ATTN: Dr. Ralph Zirkind
Westgate Research Park
McLean, VA 22101

Photon Research Associates
ATTN: Dr. J. Myer
P. O. Box 1318
La Jolla, CA 92307

Varo, Inc.
Texas Division
ATTN: Dr. K. Sewell
2201 W. Walnut Street
P. O. Box 401267
Garland, TX 75040

Lockheed Missile and Space Co.
ATTN: Mr. W. Rudolf
3251 Hanover Street
Palo Alto, CA 94304

Brunswick Corporation
Defense Division
ATTN: C. Green
2000 Brunswick Lane
Deland, FL 32720

Aerospace Corporation
El Segundo Operations
ATTN: Dr. F. Simmons/Dr. Allan Klinger
P. O. Box 95085
Los Angeles, CA 90045

IRT Corporation
ATTN: Dr. J. T. Neu
7650 Convoy Court
P. O. Box 80817
San Diego, CA 92138

Rockwell International Corporation
Space Division
ATTN: Mr. R. Arras
12214 S. Lakewood Blvd.
Downey, CA 90241

PRIVATE INDUSTRY - NONPROFIT

Institute for Defense Analysis
ATTN: Dr. L. Biberman (IDA/STD)
400 Army-Navy Drive
Arlington, VA 22202

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. T. John
P. O. Box 73
Lexington, MA 02173

The Rand Corporation
ATTN: Dr. L. S. Mundie
1700 Main Street
Santa Monica, CA 90406

Institute for Defense Analysis
ATTN: Dr. H. Wolfhard
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analysis
ATTN: Dr. E. Bauer
400 Army-Navy Drive
Arlington, VA 22202

DATE
FILMED
-8